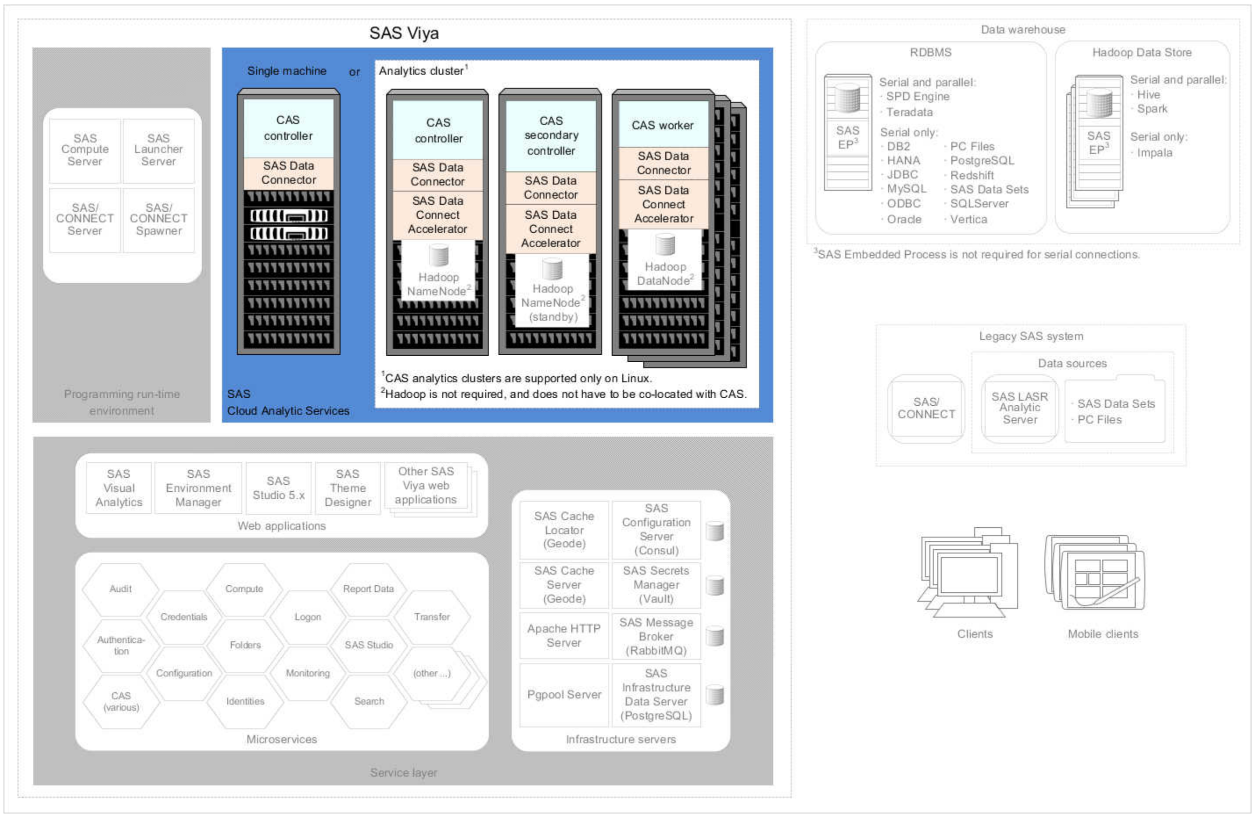
SAS® Viya® 3.5 Administration: SAS® Cloud Analytic Services

SAS Cloud Analytic Services: Overview

SAS Cloud Analytic Services (CAS) is a server that provides the cloud-based, run-time environment for data management and analytics with SAS. Suitable for both on-premises and Cloud deployments, CAS uses a combination of hardware and software where data management and analytics take place on either a single machine or as a distributed server across multiple machines.

The following diagram shows the relationship of CAS to other components in the SAS Viya full deployment:

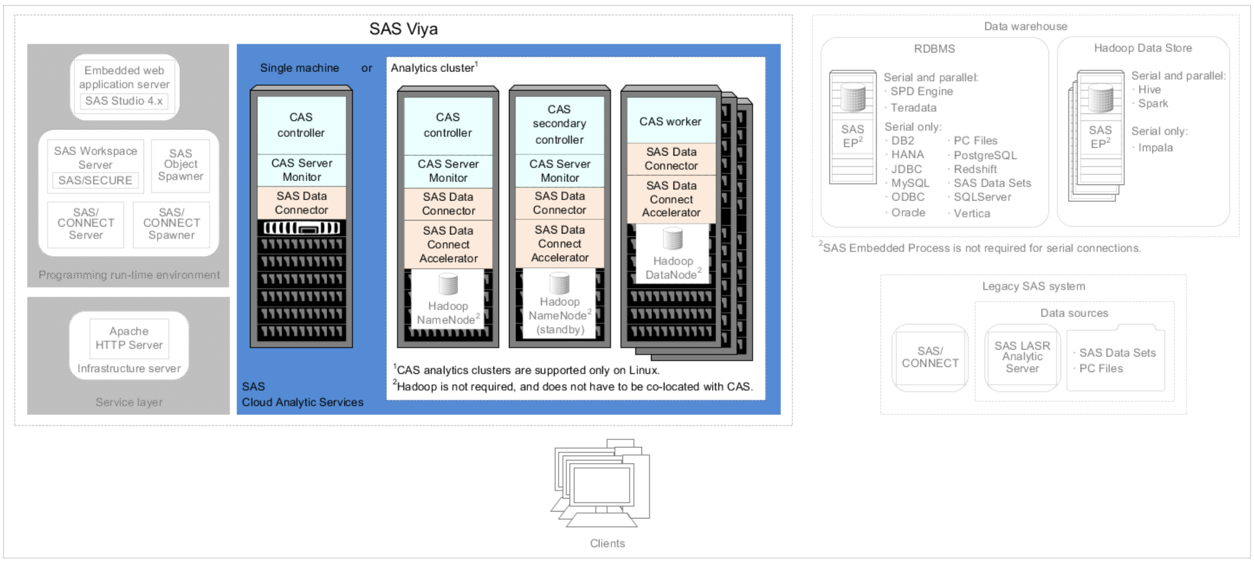
Figure 1 *SAS Cloud Analytic Services (Full Deployment)*



Note: CAS analytics clusters are supported only on Linux.

The following diagram shows the relationship of CAS to other components in the SAS Viya programming-only deployment:

Figure 2 SAS Cloud Analytic Services (Programming-only Deployment)



SAS Cloud Analytic Services: How To (Scripts)

Operate (Linux)

SAS Viya provides a script in **`/etc/init.d`** that you use to stop, start, restart, and check the status of a CAS server. The script is named, `sas-viya-cascontroller-default`.

Syntax

How you run `sas-viya-cascontroller-default` depends on your operating system:

- Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

  ```
  sudo systemctl status | stop | start | restart sas-viya-cascontroller-default
  ```

- Red Hat Enterprise Linux 6.x (or an equivalent distribution):

  ```
  sudo service sas-viya-cascontroller-default status | stop | start | restart
  ```

Usage Notes and Tips

- You must be signed in to the machine where the CAS controller resides and you must have root-level privileges to run this script. Running sas-viya-cascontroller-default affects all worker nodes.
- `sas-viya-cascontroller-default` checks the status of the CAS controller only. To check the status of an individual worker node, use SAS Environment Manager or CAS Server Monitor.
- On multi-tenant SAS Viya systems, the script is named `sas-`*`tenant-ID`*`-sas-viya-cascontroller`.

- Your site's Linux administrator might want to create a regular account (for example, sas-service-admin) and give that account the sudo permissions to manage the SAS services. Make sure that the services are defined as "start on reboot" so that the CAS server automatically starts when the machine is rebooted.
- There is a script with which you can manage and view the running state of all SAS Viya services. For more information, see "Start and Stop All Servers and Services" in *SAS Viya Administration: General Servers and Services*.
- On Linux systems that support systemd, use the `systemctl` command when running `sas-viya-cascontroller-default`. The `systemctl` command maintains a record of service status that the `service` command and a direct call does not use.

CAUTION
On Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x, do not mix System V init and systemd commands. Mixing the System V init (`service` command) with the systemd (`systemctl` command) causes several issues. The `systemctl` command knows nothing about a SAS Viya service started with the `service` command. If you start `sas-viya-cascontroller-default` on RHEL 7.x with the `service` command, and later attempt to shut down the CAS server using the `systemctl` command, the CAS server stops responding and does not shut down.

Examples

- To check status of the CAS controller on Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

```
sudo systemctl status sas-viya-cascontroller-default
```

- To stop the CAS controller and its worker nodes on Red Hat Enterprise Linux 6.x (or an equivalent distribution):

```
sudo service sas-viya-cascontroller-default stop
```

- To start the CAS controller and its worker nodes on Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

```
sudo systemctl start sas-viya-cascontroller-default
```

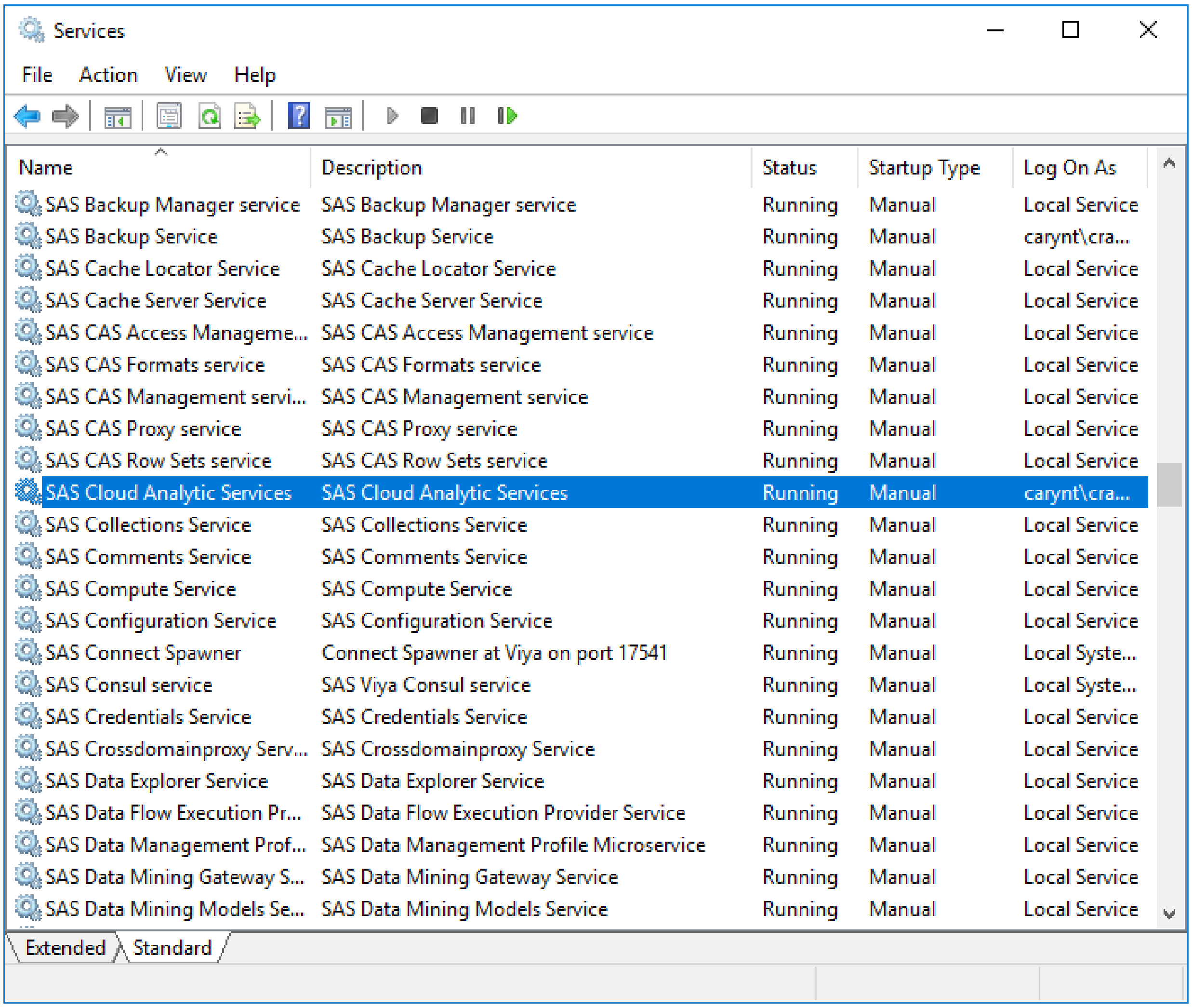
- To restart the CAS controller and its worker nodes on Red Hat Enterprise Linux 6.x (or an equivalent distribution):

```
sudo service sas-viya-cascontroller-default restart
```

Operate (Windows)

Using the Services snap-in in the Microsoft Management Console, you can start, stop, and restart SAS Cloud Analytic Services.

Figure 3 SAS Cloud Analytic Services in the Services Snap-in



Name	Description	Status	Startup Type	Log On As
SAS Backup Manager service	SAS Backup Manager service	Running	Manual	Local Service
SAS Backup Service	SAS Backup Service	Running	Manual	carynt\cra...
SAS Cache Locator Service	SAS Cache Locator Service	Running	Manual	Local Service
SAS Cache Server Service	SAS Cache Server Service	Running	Manual	Local Service
SAS CAS Access Manageme...	SAS CAS Access Management service	Running	Manual	Local Service
SAS CAS Formats service	SAS CAS Formats service	Running	Manual	Local Service
SAS CAS Management servi...	SAS CAS Management service	Running	Manual	Local Service
SAS CAS Proxy service	SAS CAS Proxy service	Running	Manual	Local Service
SAS CAS Row Sets service	SAS CAS Row Sets service	Running	Manual	Local Service
SAS Cloud Analytic Services	SAS Cloud Analytic Services	Running	Manual	carynt\cra...
SAS Collections Service	SAS Collections Service	Running	Manual	Local Service
SAS Comments Service	SAS Comments Service	Running	Manual	Local Service
SAS Compute Service	SAS Compute Service	Running	Manual	Local Service
SAS Configuration Service	SAS Configuration Service	Running	Manual	Local Service
SAS Connect Spawner	Connect Spawner at Viya on port 17541	Running	Manual	Local Syste...
SAS Consul service	SAS Viya Consul service	Running	Manual	Local Syste...
SAS Credentials Service	SAS Credentials Service	Running	Manual	Local Service
SAS Crossdomainproxy Serv...	SAS Crossdomainproxy Service	Running	Manual	Local Service
SAS Data Explorer Service	SAS Data Explorer Service	Running	Manual	Local Service
SAS Data Flow Execution Pr...	SAS Data Flow Execution Provider Service	Running	Manual	Local Service
SAS Data Management Prof...	SAS Data Management Profile Microservice	Running	Manual	Local Service
SAS Data Mining Gateway S...	SAS Data Mining Gateway Service	Running	Manual	Local Service
SAS Data Mining Models Se...	SAS Data Mining Models Service	Running	Manual	Local Service

Because there is a particular sequence in which the servers and services must be started and stopped, the individual services are not configured to run automatically when the SAS Viya machine is booted.

IMPORTANT SAS Configuration Service (Consul), SAS Infrastructure Data Server (PostgreSQL), SAS HTTP Proxy Server (Apache HTTP Server), and SAS Message Broker (RabbitMQ) are dependencies for the other SAS Viya services. If you are operating one or more services individually, always start each of these four services first and stop them last.

Note: There is one service, SAS Services Manager, that you can use to start and stop all SAS Viya servers and services. SAS Services Manager recognizes the dependencies between services and starts and stops services in the correct sequence. For more information, see "Start and Stop All Servers and Services" in *SAS Viya Administration: General Servers and Services*.

Change the Process Owner Account

1 Log on to the CAS controller machine as the SAS install user (sas) or with root-level privileges.

2 Using a text editor, open **`/opt/sas/viya/config/etc/sysconfig/cas/default/sas-cas-usermods`**.

3 Locate or add the following lines:

```
SASUSER="user-account"
SASGROUP="primary-group"
```

Note: The default process owner account is cas. The default primary group for cas is sas.

Enter the new CAS process owner account. If needed, enter a new primary group for the CAS process owner, and save the file.

4 Open **`/opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig-viya-default`**.

5 Locate the following line:

```
restrictServerLaunch=user-account
```

Enter the new CAS process owner account, and save the file.

CAS uses the new process owner account the next time it is run.

Add a Backup Controller

Follow these steps to add a backup controller to your CAS server:

1 Make sure that both the backup controller and the CAS controller (the primary controller) use the same shared file system.

When starting CAS sessions from SAS Studio or from any interface by users in the `CASHostAccountRequired` group, the users' home directories (`$HOME`) must be shared so that they can be accessed on both the controller machine and the backup controller machine. Sharing users' home directories ensures that the path for the `CASUSER` library is available during CAS session start-up.

For most other CAS session scenarios, the `CASUSER` library is set to a path in the shared file system described in "Set Up a Shared File System for CAS Controllers (Post-Deployment)".

2 Make sure that the CAS Backup Controller machine that is added to the deployment has the host name that you expect.

For more information, see "Confirm the Identities of the Hosts" in *SAS Viya for Linux: Deployment Guide*.

3 When adding to an existing SAS Viya deployment, SAS downloads and installs the latest software available from the software repositories. Therefore, make sure that you are using a mirror repository.

For more information, see “Create a Mirror Repository” in *SAS Viya for Linux: Deployment Guide*.

4 Every machine on which you are installing a backup controller must have the CAS user account (cas) and group (sas) set up.

For more information, see “Set Up the cas Account” in *SAS Viya for Linux: Deployment Guide*.

5 Sign in to the Ansible controller as the user account that deploys the software.

For more information, see “Set Up the User Account that Deploys the Software” in *SAS Viya for Linux: Deployment Guide*.

6 Update the inventory.ini file. In the inventory file, define the machines on which you are adding the backup controller.

TIP If you used the recommended location for uncompressing your playbook, the file is located at **`/sas/install/sas_viya_playbook/inventory.ini`**.

Here is an example of adding a backup controller:

```
controller_02 ansible_host=controller_02.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
```

For more information, see “Specify the Machines in the Deployment” in *SAS Viya for Linux: Deployment Guide*.

Also, in the inventory file, add the machines that you are adding to the appropriate group. Use the `sas_casserver_secondary` group.

Note: If your inventory file does not contain a `sas_casserver_secondary` group, then create one using the example that follows as a guide.

In this example, a backup controller is being added to the controller_02 machine:

```
[sas_casserver_secondary]
controller_02
```

7 Run Ansible, using the **`site.yml`** playbook.

```
ansible-playbook -i inventory.ini site.yml
```

Note: When you run site.yml playbook, Ansible stops and restarts the SAS Viya deployment.

Add a New Worker Node

Make sure that you are licensed for the additional nodes that you are planning to add to your analytic cluster.

Perform these steps to add new worker nodes to your existing MPP CAS server:

1 When adding to an existing SAS Viya deployment, SAS downloads and installs the latest software available from the software repositories. Therefore, make sure that you are using a mirror repository.

For more information, see "Create a Mirror Repository" in *SAS Viya for Linux: Deployment Guide*.

2 Every machine on which you are installing CAS worker nodes must have the CAS user account (cas) and group (sas) set up.

For more information, see "Set Up the cas Account" in *SAS Viya for Linux: Deployment Guide*.

3 Make sure that the worker node that is added to the CAS server has the host name that you expect.

For more information, see "Confirm the Identities of the Hosts" in *SAS Viya for Linux: Deployment Guide*.

4 Sign in to the Ansible controller as the user account that deploys the software.

For more information, see "Set Up the User Account that Deploys the Software" in *SAS Viya for Linux: Deployment Guide*.

5 Update the inventory.ini file. In the inventory file, define the machines on which you are adding the worker nodes.

Here is an example of adding a worker node:

```
worker_023 ansible_host=worker_23.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
```

For more information, see "Specify the Machines in the Deployment" in *SAS Viya for Linux: Deployment Guide*.

Use the `sas_casserver_worker` group.

Note: If your inventory file does not contain a `sas_casserver_worker` group, then create one using the example that follows as a guide.

In this example, a worker node is being added to the worker_23 machine:

```
[sas_casserver_worker]
 worker_019
 worker_020
 worker_021
 worker_022
 worker_023
```

6 Run Ansible, using the **`deploy-casworker.yml`** playbook.

```
ansible-playbook -i inventory.ini deploy-casworker.yml
```

When you run the deploy-casworker.yml playbook, Ansible does not restart the CAS server.

The deploy-casworker playbook installs SAS Viya and CAS software components on one or more identified hosts. If you have run the deploy-casworker playbook and want to immediately start the worker nodes that you have added, sign in to SAS Environment Manager as an administrator with superuser role privileges, and then join the new node to the MPP CAS server to make it an active participant. For instructions on how to add new worker nodes using SAS Environment Manager, see "Manage CAS Server Nodes" on page 24.

Note: If you want to add a new worker node using CAS Server Monitor, then enable and sign in to CAS Server Monitor. See "View and Manage CAS Nodes" on page 30 to add one or more new worker nodes.

Note: For more information about the deploy-casworker.yml playbook, see "Playbooks for Adding Worker Nodes" on page 52.

Add a CAS Server

To add a CAS server, follow these steps:

Note: If you want to add only a CAS worker node, or a CAS backup controller, refer to "Convert SMP CAS Server to MPP CAS Server ".

Note: In a programming-only deployment, make sure to take a backup of your proxy.conf file. After you complete all the steps to add a CAS server, add the information from the backed-up proxy.conf file to the new proxy.conf file.

1. Before you proceed, make sure that your SAS Viya system meets the requirements.

2. When adding to an existing SAS Viya deployment, SAS downloads and installs the latest software available from the software repositories. Therefore, make sure that you are using an existing mirror repository.

   For more information, see "Create a Mirror Repository" in *SAS Viya for Linux: Deployment Guide*.

   IMPORTANT If you did not use a mirror repository to deploy SAS Viya, contact SAS Technical Support before proceeding.

3. Make sure that the CAS server that is added to the deployment has the host name that you expect.

   For more information, see "Confirm the Identities of the Hosts" in *SAS Viya for Linux: Deployment Guide*.

4. Sign in to the Ansible controller as the user account that deploys the software.

   For more information, see "Set Up the User Account that Deploys the Software" in *SAS Viya for Linux: Deployment Guide*.

5. Each CAS server that you are adding must have its own vars.yml and inventory.ini files.

   IMPORTANT The recommended method for managing multiple instances of vars.yml and inventory.ini is to create a copy of each. Include the name of the CAS server in the file names of these copies. Store these new copies in the playbook directory. The examples used in this document are casserv02.vars.yml and casserv02.inventory.ini.

   Change to your Ansible playbook directory and make a copy of vars.yml.

TIP If you used the recommended location for uncompressing your playbook, vars.yml is located in **`/sas/install/sas_viya_playbook/`**.

Here is an example:

```
cp vars.yml casserv02.vars.yml
```

6 Create a unique instance name for the CAS server that you are adding.

Open the copy of vars.yml (in this document, casserv02.vars.yml) and locate the `CAS CONFIGURATION` section. Under `casenv_user`, add the following line:

`casenv_instance:`*`new-CAS-server-name`*

IMPORTANT CAS server instance names must contain only alphanumeric characters. Any case is allowed. Instance names must not contain any special characters.

Here is an example:

```
############################################################################
## CAS Configuration
############################################################################

# The user that the CAS process will run under

casenv_user: cas

 casenv_instance:server02

# The group that the CAS user belongs to

casenv_group: sas
```

TIP The value of `casenv_instance` is appended to the base name, `cas-shared`, to form the deployment instance name for the new CAS server. In this example, the full deployment instance name is: `cas-shared-server02`.

7 Evaluate each CAS server configuration option in the `CAS CONFIGURATION` section, and update each as appropriate for the new CAS server instance.

For more information, see "Configuration File Options" on page 64.

8 Also, each CAS server that you are adding must have its own inventory.ini file. Make a copy of your inventory.ini file.

Here is an example:

```
cp inventory.ini casserv02.inventory.ini
```

9 In a text editor, open the copy of the new inventory.ini file (casserv02.inventory.ini), and do the following:

a At the very top of the file, there is a list of deployment targets. Do not remove this list. (You add to this list later in this document.)

```
controller_01 ansible_host=controller_101.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
controller_02 ansible_host=controller_102.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_01 ansible_host=worker_101.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_02 ansible_host=worker_102.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_03 ansible_host=worker_103.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_04 ansible_host=worker_104.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_05 ansible_host=worker_105.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
```

b Remove the original deployment targets from all host groups in the file, except for `[consul]`, `[httpproxy]`, and `[sas_all:children]`. These three host groups must contain their original entries.

> **IMPORTANT** A host group can have no entries under it. But, a host group should not be removed, even if it is empty. Do not modify `[consul]`, `[httpproxy]`, and `[sas_all:children]` or deployment failures are likely to occur.

10 If you are adding a distributed CAS server across multiple machines, skip to Step 13.

11 If you adding a CAS non-distributed server to a single machine, perform these steps:

a Using a text editor open the copy of your inventory.ini (in this document, casserv02.inventory.ini). At the very top of the file, map a deployment target to a machine, an Ansible user, and a private key file.

Here is an example:

```
new_controller ansible_host=controller_201.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
```

b Assign the deployment targets that you mapped in Step 11a to the `[sas_casserver_primary]` host group.

Here is an example:

```
[sas_casserver_primary]
new_controller
```

c Verify that the `[consul]`, `[httpproxy]` and `[sas_all:children]` host groups are present, and that they contain the entries from your original SAS Viya deployment.

Here is an example:

```
[consul]
original_deployment_target

[httpproxy]
original_deployment_target

[sas_all:children]
hostgroup1
hostgroup2
hostgroup3
hostgroup4
```

```
.
.
.
```

12 Skip to Step 14.

13 If you are adding a distributed CAS server across multiple machines, perform these steps:

a Using a text editor open the copy of your inventory.ini (in this document, casserv02.inventory.ini). At the very top of the file, map a deployment target to a machine, an Ansible user, and a private key file.

Here is an example for mapping five deployment targets: one controller, one backup controller (optional), and three workers (highlighted text that follows):

```
controller_01 ansible_host=controller_101.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
controller_02 ansible_host=controller_102.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_01 ansible_host=worker_101.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_02 ansible_host=worker_102.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_03 ansible_host=worker_103.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_04 ansible_host=worker_104.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
worker_05 ansible_host=worker_105.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
new_controller_01 ansible_host=controller_201.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
new_controller_02 ansible_host=controller_202.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
new_worker_03 ansible_host=worker_203.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
new_worker_04 ansible_host=worker_204.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
new_worker_05 ansible_host=worker_205.example.com ansible_user=user1
ansible_ssh_private_key_file= ~/.ssh/id_rsa
```

b Assign the deployment targets that you mapped in Step 13a to the required host groups.

Here is an example:

```
[sas_casserver_primary]
new_controller_01

[sas_casserver_secondary]
new_controller_02

[sas_casserver_worker]
new_worker_03
new_worker_04
new_worker_05
```

c Verify that the `[consul]`, `[httpproxy]` and `[sas_-all:children]` host groups are present, and that they contain the entries from your original SAS Viya deployment.

Here is an example:

```
[consul]
```

```
original_deployment_target

[httpproxy]
original_deployment_target

[sas_all:children]
hostgroup1
hostgroup2
hostgroup3
hostgroup4
.
.
.
```

14 Run Ansible:

`ansible-playbook -i` ***`CAS-server-inventory-file-name`*** **`site.yml -e "@`*****`CAS-server-vars-file-name`*****`"`**

Here is an example:

```
ansible-playbook -i casserv02.inventory.ini site.yml -e "@casserv02.vars.yml"
```

For more information, see “Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4” in *SAS Viya for Linux: Deployment Guide*.

15 Verify that the CAS server that you added is running by launching gridmon.sh.

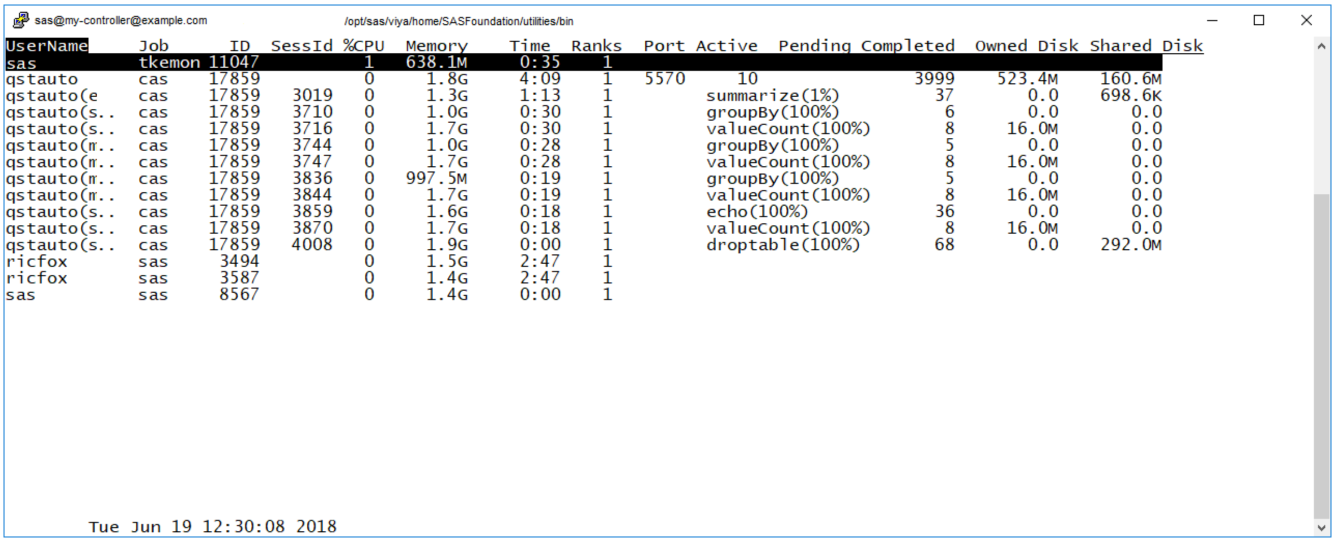
a Log on to the CAS controller machine as a user with passwordless SSH access to all CAS nodes.

b Run the following command to start gridmon.sh:

```
/opt/sas/viya/home/SASFoundation/utilities/bin/gridmon.sh
```

c You should see one or more jobs running.

Figure 4 gridmon.sh Running in Job Mode



```
sas@my-controller@example.com                 /opt/sas/viya/home/SASFoundation/utilities/bin
UserName     Job        ID  SessId %CPU  Memory     Time  Ranks  Port Active   Pending Completed  Owned Disk Shared Disk
sas          tkemon 11047            1   638.1M     0:35    1
qstauto      cas     17859           0     1.8G     4:09    1   5570      10                3999    523.4M      160.6M
qstauto(e    cas     17859    3019   0     1.3G     1:13    1         summarize(1%)           37       0.0      698.6K
qstauto(s..  cas     17859    3710   0     1.0G     0:30    1         groupBy(100%)            6       0.0         0.0
qstauto(s..  cas     17859    3716   0     1.7G     0:30    1         valueCount(100%)         8     16.0M         0.0
qstauto(m..  cas     17859    3744   0     1.0G     0:28    1         groupBy(100%)            5       0.0         0.0
qstauto(m..  cas     17859    3747   0     1.7G     0:28    1         valueCount(100%)         8     16.0M         0.0
qstauto(m..  cas     17859    3836   0   997.5M     0:19    1         groupBy(100%)            5       0.0         0.0
qstauto(m..  cas     17859    3844   0     1.7G     0:19    1         valueCount(100%)         8     16.0M         0.0
qstauto(s..  cas     17859    3859   0     1.6G     0:18    1         echo(100%)              36       0.0         0.0
qstauto(s..  cas     17859    3870   0     1.7G     0:18    1         valueCount(100%)         8     16.0M         0.0
qstauto(s..  cas     17859    4008   0     1.9G     0:00    1         droptable(100%)         68       0.0      292.0M
ricfox       sas      3494           0     1.5G     2:47    1
ricfox       sas      3587           0     1.4G     2:47    1
sas          sas      8567           0     1.4G     0:00    1

        Tue Jun 19 12:30:08 2018
```

Set Up a Shared File System for CAS Controllers (Post-Deployment)

If you want to make your CAS controller fault tolerant, during installation you can choose to deploy a CAS backup controller (also referred to as a secondary controller). A requirement for operating CAS with a backup controller is that it and the CAS controller (the primary controller) must both use the same shared file system.

1 Shut down the CAS controller.

2 Copy all of the data from **`/opt/sas/viya/config/data/cas`** to a network share that both the CAS controller and its backup controller can access.

   Note: Make sure that the copy preserves directory ownership and permissions.

   Here is an example:

```
cp -Rp /opt/sas/viya/config/data/cas /share
```

   TIP After you copy **`/opt/sas/viya/config/data/cas`**, remember that you will have a **`cas`** directory under your target. For example, if you copy **`/opt/sas/viya/config/data/cas`** to **`/share`**, you will have a cas directory under **`/share`** (**`/share/cas/`**).

3 Verify that all files and directories have been copied.

4 On the CAS controller, delete the old data directory.

   Here is an example:

```
rm -r /opt/sas/viya/config/data/cas
```

5 On both the CAS controller and the backup controller machines, create a Linux symbolic link at **`/opt/sas/viya/config/data/cas`** that points to the new shared file system.

   Here is an example:

```
ln -sf /share/cas /opt/sas/viya/config/data/cas
```

6 Start the CAS controller.

Convert SMP CAS Server to MPP CAS Server

The CAS server runs in SMP mode when it exists on a single host machine. When a single CAS server runs distributed across multiple hosts (with a controller and multiple worker hosts), it is running in MPP mode.

Perform these steps to convert a single-machine SMP CAS server to run distributed across multiple hosts in MPP mode:

1 When adding to an existing SAS Viya deployment, SAS downloads and installs the latest software available from the software repositories. Therefore, make sure that you are using a mirror repository.

   For more information, see "Create a Mirror Repository" in *SAS Viya for Linux: Deployment Guide*.

2 Every machine on which you are installing CAS worker nodes must have the CAS user account (cas) and group (sas) set up.

   For more information, see "Set Up the cas Account" in *SAS Viya for Linux: Deployment Guide*.

3 Sign in to the Ansible controller as the user account that deploys the software.

   For more information, see "Set Up the User Account that Deploys the Software" in *SAS Viya for Linux: Deployment Guide*.

4 Update the inventory.ini file. In the inventory file, define the machines on which you are adding the worker nodes.

   Here is an example of adding worker nodes:

```
casworker_001 ansible_host=worker_01.example.com ansible_user=user1
    ansible_ssh_private_key_file= ~/.ssh/id_rsa
casworker_002 ansible_host=worker_02.example.com ansible_user=user1
    ansible_ssh_private_key_file= ~/.ssh/id_rsa
casworker_003 ansible_host=worker_03.example.com ansible_user=user1
    ansible_ssh_private_key_file= ~/.ssh/id_rsa
```

   For more information, see "Specify the Machines in the Deployment" in *SAS Viya for Linux: Deployment Guide*.

   Use the `sas_casserver_worker` group.

   Note: If your inventory file does not contain a `sas_casserver_worker` group, then create one using the example that follows as a guide.

   In this example, the original SMP CAS server running on host casserver_001 is converted to act as the controller of the MPP CAS server by adding new worker nodes to the worker_01, worker_02, and worker_03 machines:

```
[sas_casserver_primary]
casserver_001

[sas_casserver_secondary]

[sas_casserver_worker]
 casworker_001
 casworker_002
casworker_003
```

5 Run Ansible, using the **`site.yml`** playbook.

```
ansible-playbook -i inventory.ini site.yml
```

   When you run the site.yml playbook, Ansible stops and restarts SAS Viya deployment.

   Note: For more information about site.yml playbook, see "Playbooks for Adding Worker Nodes" on page 52.

Note: If you are adding a worker node to an existing MPP CAS server, see "Add a New Worker Node" on page 8.

Convert an MPP CAS Worker to an SMP CAS Server

When you convert an SMP CAS server to an MPP CAS server there are certain scenarios where you want to create a separate single-machine SMP CAS server to isolate an activity. An example of this would be to create an SMP CAS server for a specific business unit while not adjusting licensing. In doing so, it is important to ensure that removing an MPP CAS worker does not impact the performance of the MPP CAS server.

Perform these steps to convert an MPP CAS worker to an SMP CAS server:

1 Make sure to back up your existing environment. See "Backup and Restore: Perform a Backup" in *SAS Viya Administration: Backup and Restore*.

2 When removing a CAS node from an existing SAS Viya deployment and converting it to an SMP CAS server, SAS downloads and installs the latest software available from the software repositories. Therefore, make sure that you use the existing playbook and an existing mirror repository. For more information, see "Create a Mirror Repository" in *SAS Viya for Linux: Deployment Guide*.

3 Remove the desired worker node from the existing MPP CAS server.

If you are using SAS Environment Manager, see "Manage CAS Server Nodes" on page 24. When you see the node status, it should show that the removed worker node is no longer connected and the data has been redistributed on other worker nodes.

Sign in to the Ansible controller using the user account that deployed the software. Navigate to the SAS Viya playbook and remove the worker node from the SAS Viya deployment inventory.ini file.

```
[sas_casserver_worker]
intcas02
intcas03
intcas04
intcas05
```

4 Run Ansible using the site.yml playbook to remove the worker node from the configuration.

```
ansible-playbook  -i inventory.ini site.yml
```

When you run the site.yml playbook, Ansible stops and restarts the SAS Viya deployment. The data needs to be reloaded once the MPP CAS server is running.

Verify that the host has been removed by displaying the hosts in the cas.hosts file on the controller of the MPP CAS server instance. You can also view it in SAS Environment Manager. See "View CAS Server Nodes" on page 23.

```
cat /opt/sas/viya/config/etc/cas/default/cas.hosts
```

```
intcas01.example.com controller
intcas02.example.com worker
intcas03.example.com worker
intcas04.example.com worker
```

5 Log on to the removed worker node (intcas05) and stop the services that are running.

```
sudo /etc/init.d/sas-viya-all-services stop
```

6 Create a new inventory file based on the original inventory file for the SMP CAS server. In this example, it is called intcas05.inventory.ini. The only host group with a host defined should be the CAS worker intcas05 and the CommandLine host group, if it is defined in the original inventory.

```
[CommandLine]
intcas05

[sas_casserver_worker]
intcas05

[sas_all]
CommandLine
sas_casserver_worker
```

7 Clean up the existing software and related items from the worker node by running the deploy_cleanup.yml playbook.

```
ansible-playbook -i intcas05.inventory.ini deploy-cleanup.yml
```

CAUTION
Make sure that the correct playbook is specified; otherwise, there is a risk of deleting your entire deployment.

8 Update the SMP CAS server inventory file (intcas05.inventory.ini).

Modify the intcas05.inventory.ini file to define hosts for the `[httpproxy]` and `[consul]` host groups. Move the host that was originally defined in the `[sas_casserver_worker]` host group to the `[sas_casserver_primary]` host group. Although the intcas05.inventory.ini file must contain all host groups from the original deployment, a host is assigned for only `[httpproxy]`, `[consul]`, `[CommandLine]` and `[sas_casserver_primary]`. Here is an example:

```
[CommandLine]
intcas05

[consul]
intviya01

[httpproxy]
intviya01

[sas_casserver_primary]
intcas05
```

9 Create a modified version of vars.yml.

In order to create another CAS server instance (SMP), create a new vars.yml file. In this example, intcas05.vars.yml is created. A parameter specifying a qualifier for the new CAS server instance is defined in the CAS CONFIGURATION: section of the intcas05.vars.yml file. The new parameter is `casenv_instance`.

```
# New instance of CAS
casenv_instance: smp
```

In this example, the name of the new SMP CAS server is cas-shared-smp. (You can choose any name for your CAS server.)

If your vars.yml file had originally specified the following options in the CAS_CONFIGURATION: section, and then make sure to remove them.

```
colocation: 'hdfs'
     mode: 'mpp'
```

After you run the playbook, the mode is set to `smp`.

10 Deploy the new SMP CAS server instance.

```
ansible-playbook -i intcas05.inventory.ini site.yml -e "@intcas05.vars.yml"
```

Note:

- The `-e` parameter is required to ensure that the new vars.yml file is included.
- If you get a `permission denied` error message while creating a machine certificate, re-create the staticcerts_token on the SAS Viya services host as the sas account and rerun the site.yml playbook. This results in a new SMP server instance running on the specified host.

11 Validate that the new SMP CAS server is active.

You can verify that the new SMP CAS server is active by logging in to SAS Environment Manager and viewing the CAS server nodes. See “View CAS Server Nodes” on page 23. You can also see the new SMP CAS instance (cas-shared-smp) in the drop-down menu of the **System Health** panel on the **Dashboard**.

Recover a Failed Controller

Note: While CAS is operating in the failed-over state, do not restart the primary (failed) controller service.

1 Shut down the CAS controller using SAS Environment Manager. (During a failover, the backup controller becomes the primary controller.)

For more information, see “Stop a CAS Server”.

2 Perform whatever steps necessary to either repair or replace the failed primary controller.

3 Restore the permstore directory from the backup controller to the primary controller.

For more information, see “Restore the Most Recent Permstore on Linux in the Event of a Failover” in *SAS Viya Administration: Backup and Restore*.

4 Do the following:

a Restart all CAS worker nodes in your deployment. (Do not start your CAS controller.)

b Start the CAS controller.

For more information, see “Operate (Linux)” on page 4.

c At the Linux command prompt, enter the **`sas-viya-controller-`*`deployment-instance`*** command and verify that the CAS controller is indeed running.

Here is an example on Red Hat Enterprise Linux 7.*x* (or an equivalent distribution):

```
sudo systemctl status sas-viya-cascontroller-default
```

```
 sas-viya-cascontroller-default is running

Host role in cluster:
Cluster Information:
    Tenant                  = shared
    Instance                = default
    Primary Controller      =
```

Set HTTP Proxy Environment Variables for CAS Sessions

If your CAS host requires a forward HTTP proxy for requests to resources such as S3 or ADLS, set the HTTP_PROXY and HTTPS_PROXY environment variables in **`/opt/sas/viya/config/etc/cas/default/cas_usermods.settings`** file on both CAS controller and backup controller machines, and worker nodes.

```
if [[ $CAS_OPTIONS == *session* ]]; then
 export HTTP_PROXY=http://proxy.example.com/
 export HTTPS_PROXY=http://proxy.example.com/
fi
```

IMPORTANT Setting the environment variables inside the `if` statement ensures that these options are applied only to the CAS user sessions and not the main CAS server process, which is necessary to prevent CAS server start-up failures.

SAS Cloud Analytic Services: How To (SAS Environment Manager)

Introduction

These instructions explain how to view and modify SAS Cloud Analytic Services (CAS) settings using SAS Environment Manager.

Navigation

To access the Servers page from SAS Environment Manager:

1 In the applications menu (☰), under **Administration**, select **Manage Environment**.

2 In the vertical navigation bar, click .

The tasks described in this section are performed from the Servers page and most can be performed only by SAS Administrators.

View CAS Server Properties and System Information

You can view CAS server properties (such as machine name and port) and system information (such as CAS version and build date).

Note: You can also view CAS server metrics.

1 Select the CAS server whose properties and system information that you want to view.

2 On the right side of the list, click .

3 To close the pane, in the top left corner of the pane, click ››.

View CAS Server Configuration

You can view CAS server configuration values and identify how they were set (for example, the maximum CAS table size, the location of the permstore, the HTTP port being used, and so on).

1 Select the CAS server whose configuration you want to view.

2 Click .

3 Make sure that the CAS Configuration tab is selected.

4 To return to the Servers page, in the top left corner of the window, click .

View CAS Start-up Options and Environment Variables

You can view environment variable and command-line option values used to run a CAS server.

1 Select the CAS server whose run-time environment you want to view.

2 Click 🔧.

3 Make sure that the **Nodes** tab is selected.

> TIP If the list of server nodes is not displayed, immediately above the list, click ☰.

4 In the list of server nodes, select the CAS controller, backup controller, or worker whose run-time environment you want to view.

5 Click ⋮⋮.

6 To return to:

- the list of CAS server nodes.

  Immediately above the list, click ☰.

- the Servers page.

  In the top left corner of the window, click ☰.

View CAS User Session Information

You can view information about a CAS server session such as the session name, session ID, connection state, owner of the session, and idle time of the session. If you are a member of the Superuser on page 46role and assume that role when prompted during sign-in to SAS Environment Manager, you can terminate sessions.

1 Select the CAS server whose sessions you want to view.

2 Click 🔧.

3 Make sure that the Sessions tab is selected.

> TIP To view additional details, add columns to the table. Click ⋮ and select **Manage columns**.

4 To return to the Servers page, in the top left corner of the window, click ☰.

The information that is displayed in the CAS user session is described here:

Name - Each application that connects to CAS, starts a session. The application is responsible for specifying the name that appears in the name column.

Session ID - The server manages each session by a UUID that uniquely identifies the session.

Owner - Shows the user that authenticated to CAS.

State - A client starts a session and remains connected to the session until the user terminates the application that started the session. However, the server does support maintaining a session in a disconnected state. The server maintains session-scope caslibs, in-memory tables, and so on until the disconnected session times out or the application reconnects to the session. The default time-out for a disconnected session is 60 seconds but each application can specify a longer time-out duration.

Idle Time - Shows the time since the last action ran in the session.

Actions Count - Shows the number of actions that ran in the session.

Terminate a CAS User Session

You can terminate all CAS sessions that you started. To terminate other users' CAS sessions, you must be a member of the Superuser on page 46role and assume that role when prompted during sign-in to SAS Environment Manager.

1 Select the CAS server whose session you want to terminate.

2 Click .

3 With the specified server highlighted, click .

4 If it is not already selected, select **Sessions**.

5 On the Sessions tab, select the check boxes for the sessions that you want to terminate, and click .

6 In the alert box that is displayed, confirm your selection by clicking **Terminate**.

7 In the top right of the window, click **Relinquish** to surrender the Superuser role for the specified server.

8 To return to the Servers page, in the top left corner of the window, click .

View CAS Server Nodes

You can view basic information such as machine name, connection state, and role for all of the nodes that belong to a CAS server.

1 Select the CAS server whose nodes you want to view.

2 Click .

3 Make sure that the Nodes tab is selected.

4 To return to the Servers page, in the top left corner of the window, click .

Manage CAS Server Nodes

To manage CAS server worker nodes, you must be a member of the Superuser on page 46role and assume that role when prompted during sign-in to SAS Environment Manager.

1 Select the distributed CAS server whose worker nodes you want to manage.

> **IMPORTANT** CAS servers running in SMP mode are non-distributed and do not have worker nodes.

2 Click .

The current admin user is granted permission to have temporary superuser role privileges. The user now can modify any operations of the CAS server.

3 With the specified server highlighted, click .

4 If it is not already selected, select **Nodes** to see the worker nodes for the specified CAS server.

> **TIP** If the list of server nodes does not display immediately above the list, click ☰.

5 To add or to remove a worker node, click .

6 In the Edit Nodes window, perform one of the following actions:

- Add a worker node

  Click + and enter the fully qualified domain name for the machine of the CAS worker node that you are adding.

- Remove a worker node

  Select the CAS worker node in the table, click 🗑, and, in the alert box, confirm the removal by clicking **Yes**.

  Note: The process of dropping (removing) a worker node ensures that active and backup copies of table blocks are preserved. This requires that all sessions complete their actions and pause while the blocks are moved. Long-running actions are canceled, and occasionally, a session might need to be killed so that the operation can proceed. The data movement often takes minutes.

7 To save any changes, click **Save**. Otherwise, click **Cancel**.

8 In the top right of the window, click **Relinquish** to surrender the Superuser role for the specified server.

9 To return to the Servers page, in the top left corner of the window, click ☰.

Manage CAS Role Memberships

For each CAS server, be sure to designate at least one user (other than the server's process owner) to the Superuser role. In the initial deployment, users that you add to the SAS Administrators predefined custom group have membership in the Superusers role. If you want to designate a user to the role without providing the extra privileges of SAS Administrators, follow these instructions.

Manage Direct Membership in the CAS Superuser Role

1. In the applications menu (☰), select **Administration** ⇨ **Manage Environment**. In the navigation bar, select 🗄.
2. Right-click a CAS server, and select **Assume the Superuser role**.
3. Right-click the server again, and select **Settings**.
4. In the Superuser Role Membership section of the Server Settings window, click ✎.
5. To add a member, do the following in the Select Identities window:
   a. In the left pane, select **Users**.
   b. In the left pane, click the name of a user. The user's properties are displayed in the far right pane.
   c. Click +›.
6. To remove a member, do the following in the Select Identities window:
   a. In the **Select Identities** list, click the user that you want to remove. The identity's properties are displayed in the right pane.

      Note: You cannot change or remove the account that starts the server.

   b. Click ‹–.
7. Click **OK**.
8. Click **Relinquish** in the status bar to relinquish the Superuser role.

Assume the Superuser Role

In SAS Environment Manager, you become a Superuser only after you explicitly assume that role. For example, you might assume the role to troubleshoot and resolve an access issue or to manage format libraries. To assume the Superuser role:

1. In the applications menu (☰), select **Administration** ⇨ **Manage Environment**. In the navigation bar, select 🗄.
2. In the list of servers, right-click the name of the server for which you want to assume the role, and select **Assume the Superuser role**.

The status message reminds you that you have assumed the role.

3 After you perform the task that required the role, click **Relinquish** in the status bar.

Note: Use the Superuser role only when it is required for a specific task. Be sure to relinquish the role when you are finished.

Manage Path Lists (Allowlists and Denylists)

To change CAS server path list (allowlist and denylist) settings, you must be a member of the Superuser on page 46role and assume that role when prompted during sign-in to SAS Environment Manager. For more information about how CAS manages allowlists and denylists, see "Paths List".

IMPORTANT CAS does not support denylists for caslibs on Amazon S3.

1 Select the CAS server whose allowlist or denylist you want to access.

2 Click .

3 With the specified server highlighted, click .

4 Make sure that the **Paths List** tab is selected.

5 To modify the active list, or to switch between a denylist, allowlist, or no list, on the right side of the Server Settings window, click .

   If you select the denylist or allowlist, you can add or remove paths to the list.

   Note: By default, the SAS Viya install and various configuration directories are on the denylist.

   IMPORTANT On Linux, if a denylist path is changed to a symbolic link, then the denylist should be updated using the fully resolved path.

6 To save any changes, click **Save**. Otherwise, click **Cancel**.

7 When you are finished, click **Close**.

8 In the top right of the window, click **Relinquish** to surrender the Superuser role for the specified server.

Adjust Caslib Management Privileges

To adjust caslib management privileges for a particular CAS server in SAS Environment Manager, you must be a member of the Superuser on page 46role and assume that role when prompted during sign-in to SAS Environment Manager.

1. Select the CAS server whose caslib management privileges you want to adjust.
2. Click .
3. With the specified server highlighted, click .
4. Select **Caslib Management Privileges** to view identities and their caslib privileges.
5. To modify privileges, click .
6. For the identities listed, choose to enable (or disable) the ability to add and delete session and global caslibs.

   Regardless of access controls, the Superuser can add and manage all caslibs.

   Note: This display shows directly granted privileges. Indirectly granted privileges and denials of privileges are not reflected in this display.

7. To save any changes, click **Save**. Otherwise, click **Cancel**.
8. When you are finished, click **Close**.
9. In the top right of the window, click **Relinquish** to surrender the Superuser role for the specified server.

Stop a CAS Server

To shut down a CAS server, you must be a member of the Superuser on page 46role and assume that role when prompted during sign-in to SAS Environment Manager.

1. Select the CAS server that you want to stop.
2. Click .
3. Right-click the CAS server, and select **Stop server**.
4. In the alert box that is displayed, confirm your selection by clicking **Stop the Server**.
5. In the top right of the window, click **Relinquish** to surrender the Superuser role for the specified server.

Remove CAS Worker Software

1. To remove a CAS worker, you must be a member of the Superuser on page 46 role and assume that role when prompted during sign-in to SAS Environment Manager.
2. Drop the CAS worker node whose software you want to remove.

   For more information, see “Manage CAS Server Nodes”.
3. Sign in to the CAS controller machine with root-level privileges.

4 On the CAS controller machine, remove the machine name of the worker node from `/opt/sas/viya/config/etc/cas/default/cas.hosts`.

5 Edit the inventory file on the Ansible controller machine to remove the deploy target definition at the top of the file. Also remove the deploy target name from [sas_casserver_worker].

6 Run the site.yml playbook to update Consul with the correct list of hosts.

7 Sign in to the CAS worker machine with root-level privileges.

8 Run the following commands from an operating system prompt:

```
sudo /opt/sas/viya/home/libexec/deployment/stop-all-services.sh
```

```
sudo /opt/sas/viya/home/libexec/deployment/pre-uninstall.sh
```

```
sudo /opt/sas/viya/home/libexec/deployment/uninstall.sh
```

```
sudo mv /opt/sas/viya/ /opt/sas/viya_$(date +"%m_%d_%Y")
```

SAS Cloud Analytic Services: How To (CAS Server Monitor)

View CAS Server Properties and System Information

1 Sign in to CAS Server Monitor with a valid user ID and password.

2 In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .

3 On the System State page, make sure that **Controller** is selected.

> **IMPORTANT** After a CAS license is renewed, the **License File** field is not updated until the CAS server is restarted.

View CAS Server Configuration

To use CAS Server Monitor to view the current list of CAS Server options and their values, follow these steps:

1 Sign in to CAS Server Monitor with a valid user ID and password.

2 In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .

3 On the Configuration page, make sure that **CAS Configuration** is selected.

View CAS Start-up Options and Environment Variables

You can use CAS Server Monitor to view the option used when a CAS server was started and to see the current list of CAS environment variables and their values.

To view CAS start-up options and environment variable values, follow these steps:

1 Sign in to CAS Server Monitor with a valid user ID and password.

2 In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .

3 On the System State page, select **Runtime Environment**.

View CAS User Session Information

You can use CAS Server Monitor to view information about a user’s session, such as connection port, length of connection time, and so on.

To view user session information, follow these steps:

1 Sign in to CAS Server Monitor with a valid user ID and password.

2 In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .

3 On the System State page, select **User Sessions**.

Cancel a CAS User Session

To cancel a CAS server session, follow these steps:

1 Sign in to CAS Server Monitor with a valid user ID and password.

Note: If you are canceling another user’s session, you must sign in to CAS Server Monitor with a user ID that has CAS Administrator privileges on page 46.

2 In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .

3 On the System State page, select **User Sessions**.

4 At the end of the row for the session that you want to cancel, click ⋮ and select **Cancel Session**.

TIP You can also use ⋮ to launch the Resource Monitor, cancel the CAS action, and terminate the session.

Note: When **Cancel Session** is selected, a message is sent to the session to interrupt the currently running action. If the action can interpret this message, the session ends the action so that the session is ready to accept the next action. If the action cannot interpret the message, the session cannot end the action. Therefore, the session must be terminated. See "Terminate a CAS User Session" on page 30.

Terminate a CAS User Session

To terminate your CAS server session, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .
3. On the System State page, select **User Sessions**.
4. At the end of the row for the session that you want to terminate, click ⋮ and select **Terminate Session**.

   When **Terminate Session** is selected, all the session processes are killed.

   TIP You can also use ⋮ to launch the Resource Monitor, cancel the CAS action, and terminate the session.

View and Manage CAS Nodes

You can use CAS Server Monitor to view, add, and remove CAS nodes in your analytics cluster.

Note: In order to add CAS nodes, the requisite software must already have been deployed on the machines that you are adding. To add new machines, deploy SAS on them first, and then you can add them using the CAS Server Monitor.

To view or manage a node, follow these steps:

1. Sign in to CAS Server Monitor with a user ID that has CAS Administrator privileges on page 46.

   Note: If you want only to view CAS server nodes, CAS Administrator privileges are not required.

2. In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .

3 On the System State page, select **Nodes**.

4 From the **Nodes** table, you can:

- View information about all the nodes in your analytics cluster.
- Add nodes to your analytics cluster:

  Note: Before you can add *new* nodes to your cluster, you must have already added the CAS worker node software to the machine. For more information, see "Convert SMP CAS Server to MPP CAS Server ".

  - ☐ Click **Add Nodes**.
  - ☐ On the Add Nodes window, in **Hostname**, enter a simple host name, such as mygrid011, and click **OK**. The server monitor runs the CAS addNode action, which starts (or restarts) the node and joins it to the cluster.

    Separate multiple host names with a comma.

    If your hosts are named in numeric order (for example, host002, host003, ...) you can enter a range of host names. Use the form, `host[start-number-end-number]` (for example, `mygrid[002-030]`).

- Drop worker nodes from your analytics cluster:

  Next to the node that you want to drop, click ⋮ and select **Remove Nodes**. The server monitor runs the CAS removeNode action, which stops the node and redistributes its data to other nodes in the cluster.

  Note: The process of dropping (removing) a node ensures that active and backup copies of table blocks are preserved. This requires that all sessions complete their actions and pause while the blocks are moved. Long-running actions are canceled, and occasionally, a session might need to be killed so that the operation can proceed. The data movement often takes minutes.

  TIP Removing a node is best suited for times when the system is mainly processing batch jobs, where a delay is not a concern.

- View information about processes running on a particular node:

  Next to the node that you want to view process information about, click ⋮ and select **Show Processes**.

- Stop the server immediately by sending a kill signal to the server process:

  Next to the node that you want to stop, click ⋮ and select **Terminate Server Instance**. The server monitor issues a command to kill the node process.

  Note: Terminate a server instance only after having tried removing a node. Using terminate might not release resources (for example, mapped memory and memory involving database connections, and so on).

Adjust Caslib Management Privileges

To enable non-administrators to add global caslibs:

1 Sign in to CAS Server Monitor with a valid user ID and password that has administrator privileges.

2 In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click 🔧.

3 On the Configuration page, select **Access Controls**.

4 In the **Caslibs** list, select **Global Caslib Creation**.

> TIP If the **Global Caslib Creation** caslib is not listed, you are not signed in as an administrator.

5 In the upper right, click **Edit**.

6 In the **Edit Access Controls** window, adjust values as needed.

Intent	Instructions
Enable all users to add global caslibs.	In the existing row for **Authenticated Users**, select the **Grant** radio button.
Enable a group to add global caslibs.	Click **Add Row**. Select **Group**, enter the group name, and select the **Grant** radio button.
Enable an individual user to add global caslibs.	Click **Add Row**. Select **User**, enter the user name, and select the **Grant** radio button.

7 Click **OK** to save your changes.

8 Under **Access Controls**, review the results of your changes.

9 Verify that users who should be able to add global caslibs can do so.

Here are details:

- User and group names that you enter are not validated.
- Regardless of access controls, administrators can add and manage all caslibs.
- For the special caslibs (**Global Caslib Creation** and **Session Caslib Creation**), the only available value in the **Activity** column is **Manage Access**. The special caslibs are protected by role requirements, not by the ManageAccess permission. Granting or denying the ManageAccess permission on the special caslibs affects only the ability of non-administrators to manage other caslibs.
- If you want to restrict the ability to manage session caslibs, select **Session Caslib Creation** in the **Caslibs** list. Add direct denials as needed.

Stop a CAS Server

You can use CAS Server Monitor to shut down the CAS server. For a distributed CAS server, in addition to the controller, clicking the **shutdown** button stops all the CAS worker nodes and the backup controller (if present).

1. Sign in to CAS Server Monitor with a user ID that has CAS Administrator privileges on page 46.
2. In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click .
3. On the System State page, make sure that **Controller** is selected.
4. At the top, on the right side of the page, click **Shutdown**.

Remove CAS Worker Software

1. Sign in to CAS Server Monitor with a user ID that has CAS Administrator privileges on page 46.
2. Drop the CAS worker node whose software you want to remove.

   For more information, see "View and Manage CAS Nodes" on page 30.
3. Sign in to the CAS controller machine with root-level privileges.
4. On the CAS controller machine, remove the machine name of the worker node from `/opt/sas/viya/config/etc/cas/default/cas.hosts`.
5. Edit the inventory file on the Ansible controller machine to remove the deploy target definition at the top of the file. Also remove the deploy target name from [sas_casserver_worker].
6. Run the site.yml playbook to update Consul with the correct list of hosts.
7. Sign in to the CAS worker machine with sudo or root-level privileges.
8. Run the following commands from an operating system prompt:

```
sudo /opt/sas/viya/home/libexec/deployment/stop-all-services.sh

sudo /opt/sas/viya/home/libexec/deployment/pre-uninstall.sh

sudo /opt/sas/viya/home/libexec/deployment/uninstall.sh

sudo mv /opt/sas/viya/ /opt/sas/viya_$(date +"%m_%d_%Y")
```

SAS Cloud Analytic Services: How To (gridmon.sh)

Overview

Note: gridmon.sh is supported only on Linux platforms.

gridmon.sh is a console or terminal application that can be run from a Linux terminal or a terminal emulator like PuTTY. gridmon.sh displays data streamed from all the machines on your CAS server showing information about jobs, individual machines on the server, and attached disks.

gridmon.sh enables you to perform several limited actions, such as killing a job, killing a rank, or running gstack. (For a complete list of functionality, see "gridmon.sh Commands".) If an X Server resides on the CAS controller, then you can launch an Xterm, a Perf Top, or an Attach Debugger session directly from gridmon.sh.

Note: Attach Debugger is for use only when directed by SAS Technical Support or by SAS R&D.

If you run gridmon in record mode, gridmon captures this streamed data. Using the playback feature, you can investigate the state of your CAS server while it was recorded.

Use gridmon.sh (Linux)

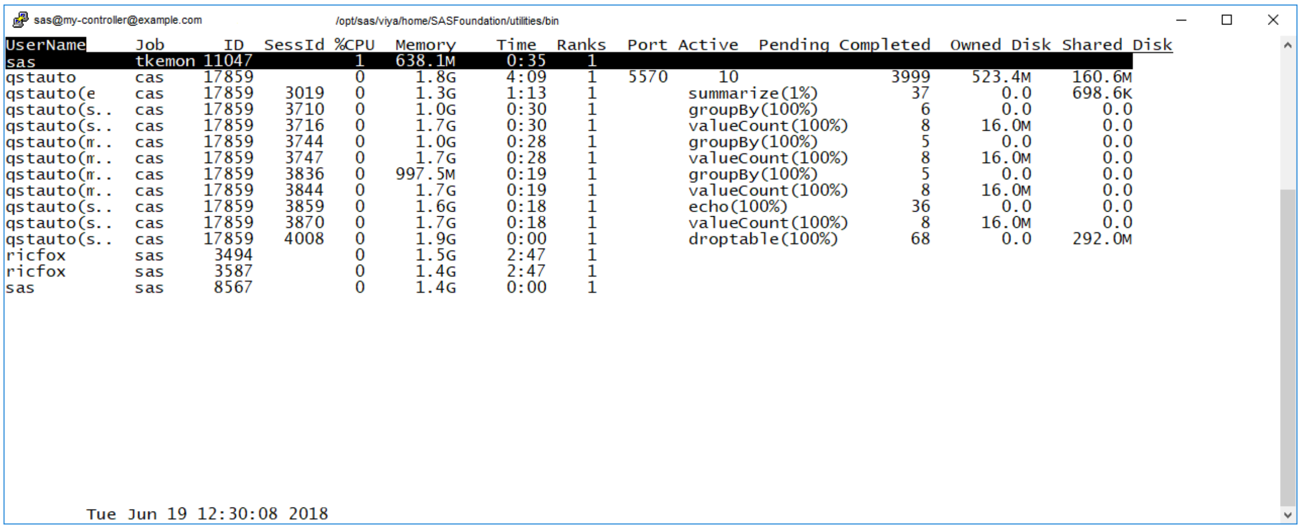
1 Log on to the CAS controller machine as a user with passwordless SSH access to all CAS nodes. The user also needs sudo privileges on all CAS nodes to run Grid Monitor commands that require root access, such as viewing process limits and killing jobs.

2 To start gridmon.sh, run the following command:

```
/opt/sas/viya/home/SASFoundation/utilities/bin/gridmon.sh
```

3 By default, gridmon.sh runs in job mode.

Figure 5 gridmon.sh Running in Job Mode

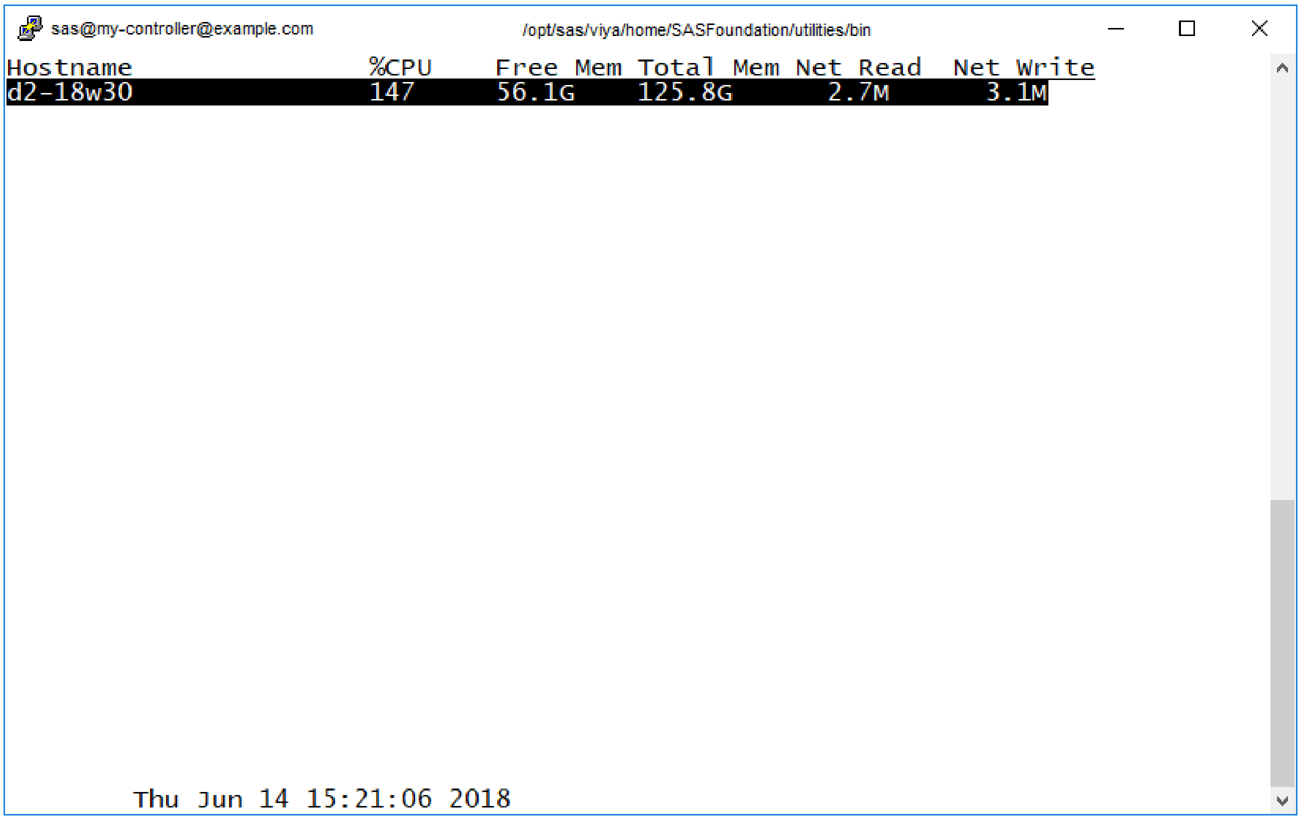


```
sas@my-controller@example.com                    /opt/sas/viya/home/SASFoundation/utilities/bin
UserName     Job      ID  SessId %CPU  Memory    Time  Ranks  Port Active  Pending Completed  Owned Disk Shared Disk
sas          tkemon 11047          1   638.1M    0:35    1
qstauto      cas    17859          0     1.8G    4:09    1  5570      10               3999    523.4M    160.6M
qstauto(e    cas    17859   3019   0     1.3G    1:13    1        summarize(1%)          37       0.0    698.6K
qstauto(s..  cas    17859   3710   0     1.0G    0:30    1        groupBy(100%)           6       0.0       0.0
qstauto(s..  cas    17859   3716   0     1.7G    0:30    1        valueCount(100%)        8     16.0M       0.0
qstauto(m..  cas    17859   3744   0     1.0G    0:28    1        groupBy(100%)           5       0.0       0.0
qstauto(m..  cas    17859   3747   0     1.7G    0:28    1        valueCount(100%)        8     16.0M       0.0
qstauto(m..  cas    17859   3836   0   997.5M    0:19    1        groupBy(100%)           5       0.0       0.0
qstauto(m..  cas    17859   3844   0     1.7G    0:19    1        valueCount(100%)        8     16.0M       0.0
qstauto(s..  cas    17859   3859   0     1.6G    0:18    1        echo(100%)             36       0.0       0.0
qstauto(s..  cas    17859   3870   0     1.7G    0:18    1        valueCount(100%)        8     16.0M       0.0
qstauto(s..  cas    17859   4008   0     1.9G    0:00    1        droptable(100%)        68       0.0    292.0M
ricfox       sas     3494          0     1.5G    2:47    1
ricfox       sas     3587          0     1.4G    2:47    1
sas          sas     8567          0     1.4G    0:00    1



        Tue Jun 19 12:30:08 2018
```

Note: **Shared Disk** consists of the sum of HDFSSize, DNFSSize, and Global FSSize across all ranks. **Owned Disk** is the sum of disk space in CAS_DISK_CACHE across all CAS worker nodes. For more information, see Table 10.

4 To run in machine mode, enter `m`.

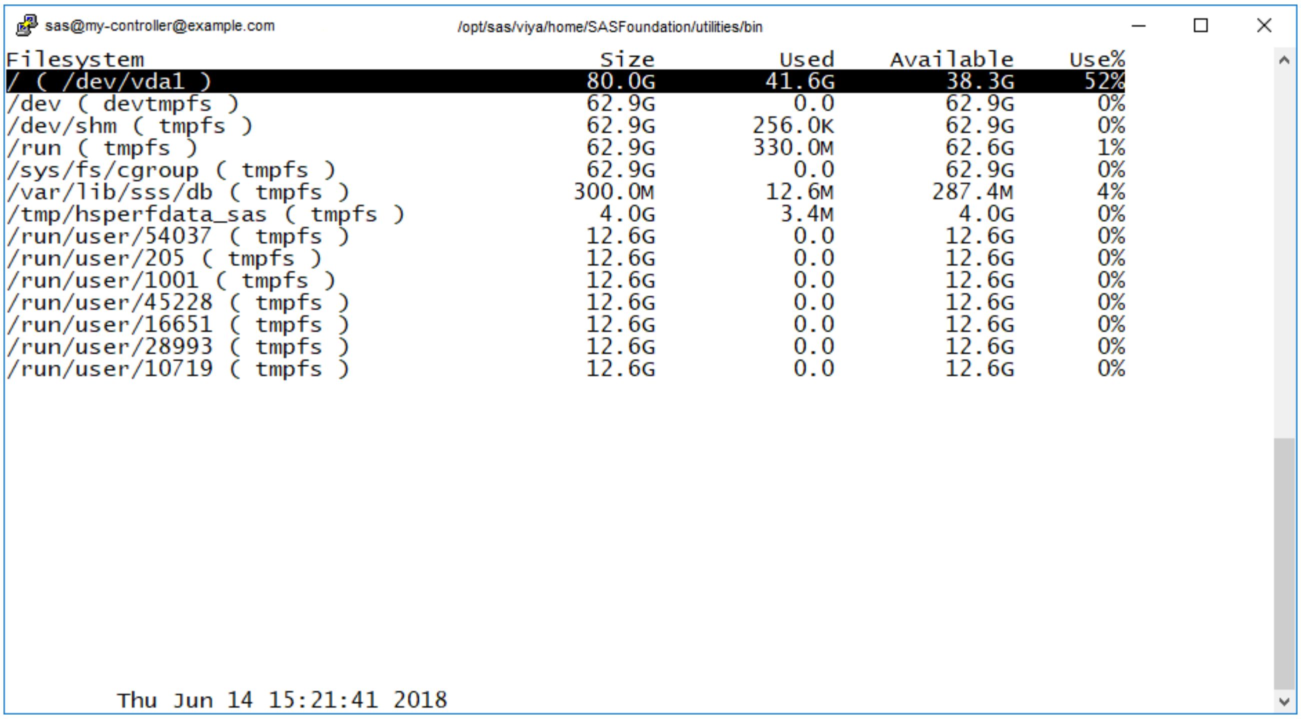
Figure 6 gridmon.sh Running in Machine Mode



```
sas@my-controller@example.com               /opt/sas/viya/home/SASFoundation/utilities/bin
Hostname                    %CPU      Free Mem  Total Mem  Net Read   Net Write
d2-18w30                    147       56.1G     125.8G          2.7M       3.1M



          Thu Jun 14 15:21:06 2018
```

5 To run in disk mode, enter `d`.

Figure 7 *gridmon.sh Running in Disk Mode*


sas@my-controller@example.com
/opt/sas/viya/home/SASFoundation/utilities/bin

```
Filesystem                                     Size         Used    Available     Use%
/ ( /dev/vda1 )                               80.0G        41.6G        38.3G      52%
/dev ( devtmpfs )                             62.9G          0.0        62.9G       0%
/dev/shm ( tmpfs )                            62.9G       256.0K        62.9G       0%
/run ( tmpfs )                                62.9G       330.0M        62.6G       1%
/sys/fs/cgroup ( tmpfs )                      62.9G          0.0        62.9G       0%
/var/lib/sss/db ( tmpfs )                    300.0M        12.6M       287.4M       4%
/tmp/hsperfdata_sas ( tmpfs )                  4.0G         3.4M         4.0G       0%
/run/user/54037 ( tmpfs )                     12.6G          0.0        12.6G       0%
/run/user/205 ( tmpfs )                       12.6G          0.0        12.6G       0%
/run/user/1001 ( tmpfs )                      12.6G          0.0        12.6G       0%
/run/user/45228 ( tmpfs )                     12.6G          0.0        12.6G       0%
/run/user/16651 ( tmpfs )                     12.6G          0.0        12.6G       0%
/run/user/28993 ( tmpfs )                     12.6G          0.0        12.6G       0%
/run/user/10719 ( tmpfs )                     12.6G          0.0        12.6G       0%

        Thu Jun 14 15:21:41 2018
```

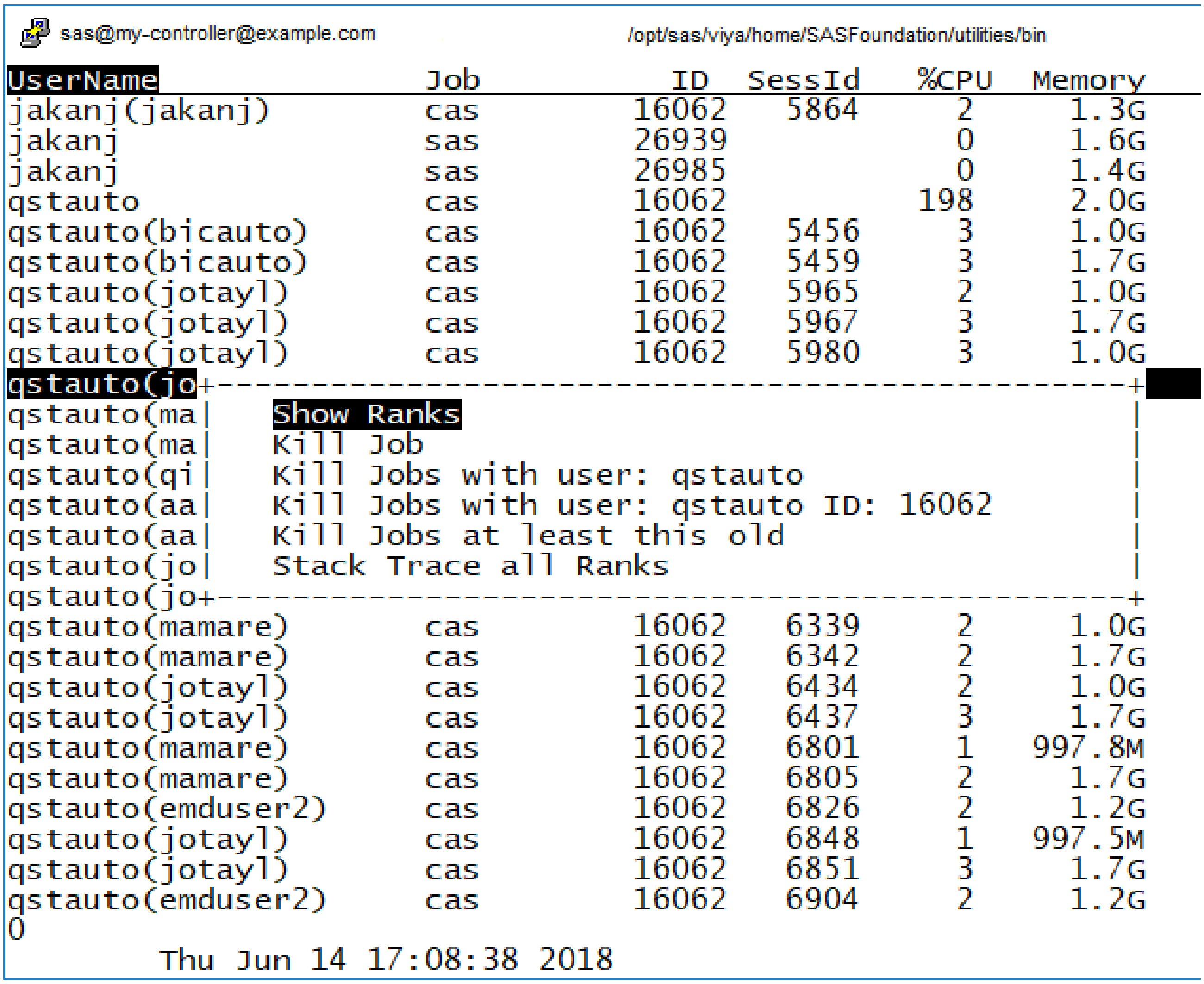
6 Refer to “gridmon.sh Commands” for the commands that you can use in each mode.

7 In job mode there are two menus.

Run in job mode (enter `j`), select a job, and press **`Enter`**.

The **Show Ranks** menu is displayed:

Figure 8 Show Ranks Menu

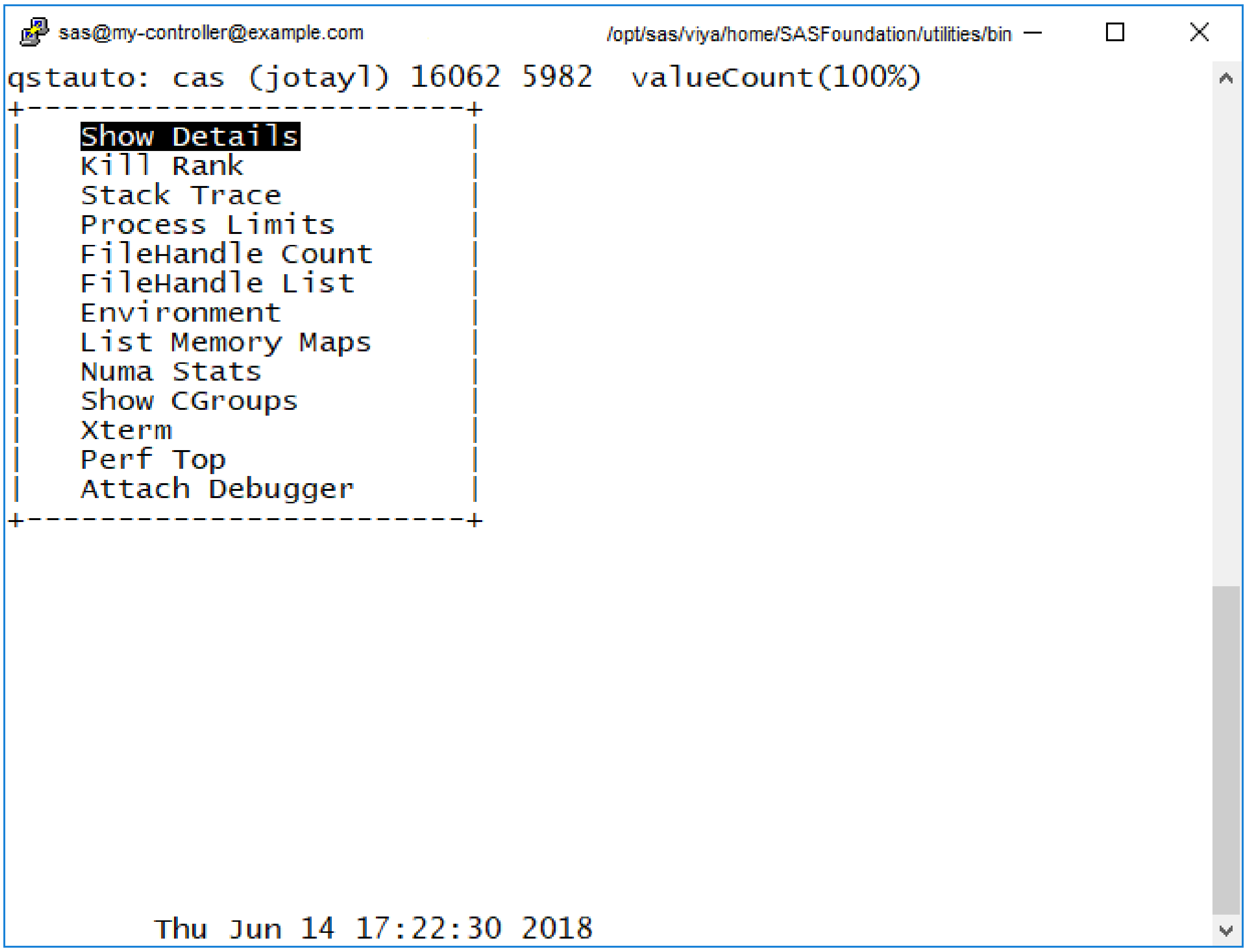


```
sas@my-controller@example.com                         /opt/sas/viya/home/SASFoundation/utilities/bin
UserName                    Job            ID    SessId    %CPU   Memory
jakanj(jakanj)              cas         16062     5864        2     1.3G
jakanj                      sas         26939                 0     1.6G
jakanj                      sas         26985                 0     1.4G
qstauto                     cas         16062               198     2.0G
qstauto(bicauto)            cas         16062     5456        3     1.0G
qstauto(bicauto)            cas         16062     5459        3     1.7G
qstauto(jotayl)             cas         16062     5965        2     1.0G
qstauto(jotayl)             cas         16062     5967        3     1.7G
qstauto(jotayl)             cas         16062     5980        3     1.0G
qstauto(jo+-------------------------------------------------------------+
qstauto(ma|    Show Ranks                                               |
qstauto(ma|    Kill Job                                                 |
qstauto(qi|    Kill Jobs with user: qstauto                             |
qstauto(aa|    Kill Jobs with user: qstauto ID: 16062                   |
qstauto(aa|    Kill Jobs at least this old                              |
qstauto(jo|    Stack Trace all Ranks                                    |
qstauto(jo+-------------------------------------------------------------+
qstauto(mamare)             cas         16062     6339        2     1.0G
qstauto(mamare)             cas         16062     6342        2     1.7G
qstauto(jotayl)             cas         16062     6434        2     1.0G
qstauto(jotayl)             cas         16062     6437        3     1.7G
qstauto(mamare)             cas         16062     6801        1   997.8M
qstauto(mamare)             cas         16062     6805        2     1.7G
qstauto(emduser2)           cas         16062     6826        2     1.2G
qstauto(jotayl)             cas         16062     6848        1   997.5M
qstauto(jotayl)             cas         16062     6851        3     1.7G
qstauto(emduser2)           cas         16062     6904        2     1.2G
0
          Thu Jun 14 17:08:38 2018
```

For specific information about each **Show Ranks** menu command, see “Show Ranks Menu Commands ”.

8 From the **Show Ranks** menu, select **Show Ranks** and the Ranks window is displayed. Press **Enter** and the **Show Details** menu is displayed.

Figure 9 *Show Details Menu*



sas@my-controller@example.com /opt/sas/viya/home/SASFoundation/utilities/bin

```
qstauto: cas (jotayl) 16062 5982  valueCount(100%)
+-------------------------+
|    Show Details         |
|    Kill Rank            |
|    Stack Trace          |
|    Process Limits       |
|    FileHandle Count     |
|    FileHandle List      |
|    Environment          |
|    List Memory Maps     |
|    Numa Stats           |
|    Show CGroups         |
|    Xterm                |
|    Perf Top             |
|    Attach Debugger      |
+-------------------------+






        Thu Jun 14 17:22:30 2018
```

For specific information about each **Show Details** menu command, see "Show Details Menu Commands ".

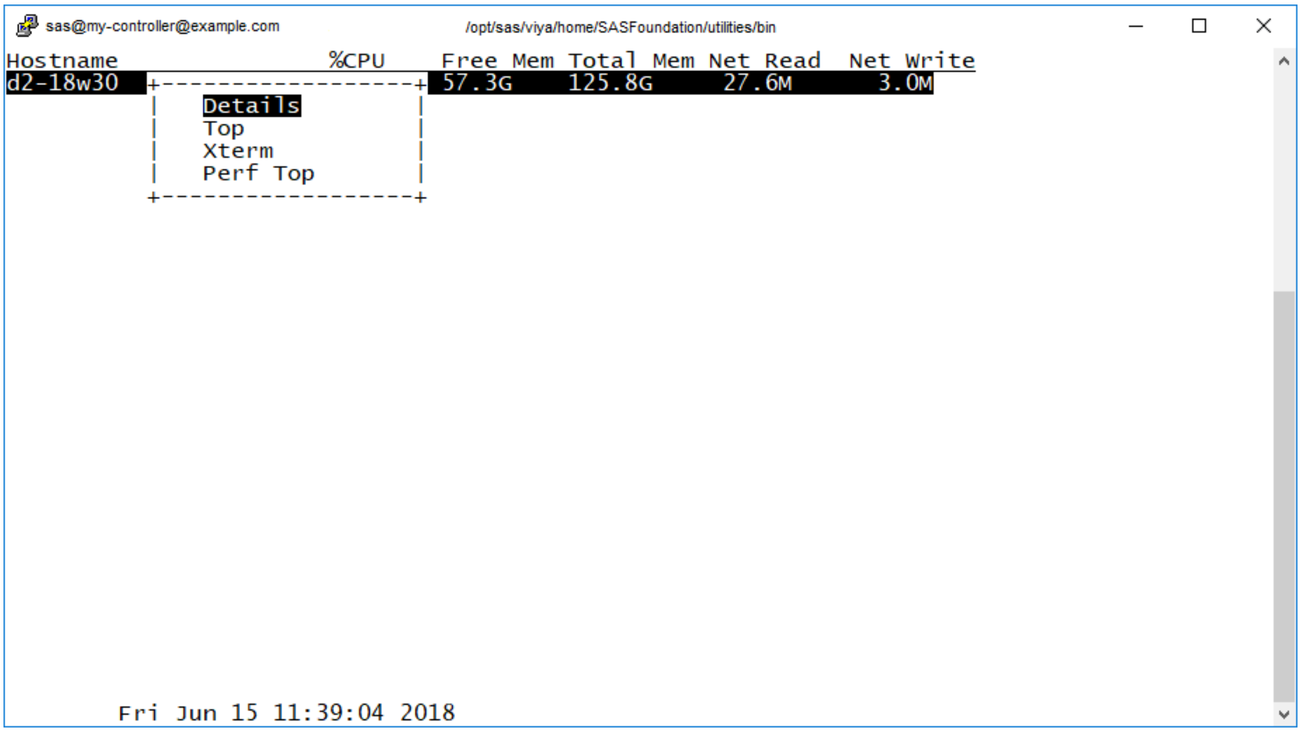
9 Press `Esc` to leave the **Show Details** menu.

10 In machine mode there is one menu.

Run in machine mode (enter `m`), select a machine and press `Enter`.

The **Details** menu is displayed:

Figure 10 Details Menu



```
sas@my-controller@example.com                /opt/sas/viya/home/SASFoundation/utilities/bin
Hostname                      %CPU     Free Mem Total Mem Net Read  Net Write
d2-18w30    +-------------------+  57.3G    125.8G     27.6M        3.0M
            |    Details        |
            |    Top            |
            |    Xterm          |
            |    Perf Top       |
            +-------------------+

          Fri Jun 15 11:39:04 2018
```

For specific information about each **Details** menu command, see "Details Menu Commands ".

11 Enter `q` to exit gridmon.sh.

Run gridmon.sh in Record Mode (Linux)

You can run gridmon.sh in record mode in order to capture data that is streamed from each machine on your CAS server at approximately one second intervals. You can review this captured data later by running gridmon.sh in playback mode.

1 Log on to the CAS controller machine as a user with passwordless SSH access to all CAS nodes. The user also needs sudo privileges on all CAS nodes to run Grid Monitor commands that require root access, such as viewing process limits and killing jobs.

2 Change to the following directory:

```
cd /opt/sas/viya/home/SASFoundation/utilities/bin/
```

3 To start gridmon.sh in record mode, run the following command:

```
./gridmon.sh -record path/output-filename
```

where ***path/output-filename*** is the absolute path and file name for where gridmon.sh writes its output.

Here is an example:

```
./gridmon.sh -record /my_data/tkgridmon_output
```

Run gridmon.sh in Playback Mode (Linux)

You can run gridmon.sh in playback mode to review data streamed from all the machines on your CAS server that you captured earlier while running gridmon.sh in record mode.

1 Log on to the CAS controller machine as a user with passwordless SSH access to all CAS nodes. The user also needs sudo privileges on all CAS nodes to run Grid Monitor commands that require root access, such as viewing process limits and killing jobs.

2 To start gridmon.sh in playback mode, run the following command:

   ```
   ./gridmon.sh -playback path/output-filename
   ```

   Here is an example:

   ```
   ./gridmon.sh -gridhost -playback /my_data/tkgridmon_output
   ```

Monitored Servers and Machines

The default hosts monitored by gridmon can be altered by creating the file **/opt/sas/viya/config/etc/cas/gridmon_usermods.hosts** where the name of the host machine and its role are declared. For example, you might want to see the following file: **/opt/sas/viya/config/etc/cas/<*instance*>/cas.hosts** on your machine. Depending on the number of hosts, the file might look similar to this:

```
<machine-name1> controller
<machine-name2> worker
<machine-name3> worker
.
.
<machine-namen> worker
```

SAS Cloud Analytic Services: How To (CLI)

The following examples assume that you have already signed in to SAS Viya at the command line. See “Command-Line Interface: Preliminary Instructions” in *SAS Viya Administration: Using the Command-Line Interfaces*.

IMPORTANT For Windows users of the CAS CLI who are using it for CAS administration and who are CAS administrators, CAS sessions are launched under the CAS service account. Also, any user who has credentials that are stored can use those credentials to run the CAS

CLI for any purpose. See “External Credentials: How To” in *SAS Viya Administration: External Credentials*.

Manage CAS Role Memberships

Example: List the administrative users on the specified CAS server.

```
sas-admin cas admin-users list --server serverA
```

Example: Add the user user1 to the Superuser role on the specified CAS server.

```
sas-admin cas admin-users add user1 --server serverA
```

Example: Remove the group with the ID group1 and name group1_name from the Superuser role on the specified CAS server. The action is performed without prompting the user for confirmation since the **`force`** option is used.

```
sas-admin cas admin-users remove --group group1 --server serverA --name group1_name --
force
```

TIP To remove a user or a group from the administrative users, you must specify the name of the user or the group as well as the identity of the user or the group. Use the name option to specify the name. Use the user option to specify the ID for a user. Use the group option to specify the ID for a group. In order to obtain the ID, use the **`admin-users list`** command.

Manage SAS Sessions

Example: List the sessions for which you are the owner on the specified CAS server.

```
sas-admin cas sessions list --server serverA
```

Example: Using elevated privileges, list 50 sessions for which the owner is user1 on the specified CAS server, and sort by **`state`**. You must be able to assume the Superuser role to run the command with elevated privileges.

```
sas-admin cas sessions list --server serverA --superuser --limit 50 --owner user1 --
sort-by state
```

Example: Using elevated privileges, list all sessions for which the name contains the string dataExplorer on the specified CAS server. You must be able to assume the Superuser role to run the command with elevated privileges.

```
sas-admin cas sessions list --server serverA --superuser --all --name-contains
dataExplorer
```

Example: Using elevated privileges, show additional information about the session with ID 12345 on the specified CAS server. You must be able to assume the Superuser role to run the command with elevated privileges.

```
sas-admin cas sessions show-info --superuser --session-id 12345  --server serverA
```

Manage Servers

Note: These examples explicitly specify the **`server`** required option when applicable. However, using an environment variable might be more efficient than this option. For more information about the environment variables, see “Details” in *SAS Viya Administration: Using the Command-Line Interfaces*.

Example: List all the CAS servers.

```
sas-admin cas servers list --all
```

Example: Adds multiple paths to the paths list on the specified server so that a caslib that is based on the paths can be created. The active list in this example is the allowlist. You can specify multiple paths in the file that is referenced in the **`source-file`** option. See caslib paths list on page 51 for more information.

```
sas-admin cas servers paths-list add-paths --server serverA --source-file
/path-to-source-file/source.txt
```

Example: On the specified server, list the identities of those who can create and delete session and global caslibs.

```
sas-admin cas servers privileges list --server serverA
```

Example: On the specified server, grant the ability to manage global caslibs to a group.

Note: A reference to the group * corresponds to Authenticated Users. To specify the Authenticated Users group, enter `--group '*'`

```
sas-admin cas servers privileges modify --server serverA --grant --global --group
group-name
```

Example: Show detailed information about the specified server. Show the dates and times in the local time zone.

```
sas-admin cas servers show-info --server serverA --use-local-datetime
```

Example: List all the resource management policies on the specified server.

```
sas-admin cas servers policies list  --all --server cas-shared-default
```

Example: Show detailed information about the priority level 1 policy on a specified server.

```
sas-admin cas servers policies show-info --policy cas-shared-default-priority-1 --
server cas-shared-default
```

Example: Create a new resource management policy for a global caslib. You can also create a global caslib policy from a JSON template.

```
sas-admin cas servers policies define global-caslibs
--attributes cpu:30,Public:1000000000,HPS:4000000000 --server serverA
```

Example: Create a new resource management policy for priority assignments. You can also create a priority assignment policy from a JSON template.

```
sas-admin cas servers policies define priority-assignments
 --attributes userA:4,userB:3 --server serverA
```

Example: Create a new resource management policy for priority level 5 on serverA. This creates a policy with the name **`serverA-priority-5`**. You can also create a priority level policy from a JSON template.

```
sas-admin cas servers policies define priority-levels --cpu 10 --global-casuser
500000000
--global-casuser-hdfs 100000000 --session-tables 2500000000 --priority 5  --server
serverA
```

Example: Create a new resource management policy for priority level 4 on serverA. Specify a source file that contains values for priority level 3, but override them with the values for priority 4 by including them on the command line. This example overrides the value for CPU in the source file by specifying the **`cpu`** option on the command line. This example assumes that you have a preexisting JSON file for a priority level 3 policy named priority3-levels.json. You can also create a priority level policy from a JSON template.

```
sas-admin cas servers policies define priority-levels --priority 4 --server serverA --
cpu 15
 --source-file /path/priority3-levels.json
```

Note:

- For CAS to use resource management policies, the environment variable **`CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT`** must be set on the CAS server. For more information, see Environment Variables.
- When you create a resource management policy for priority levels, the policy is named ***`CAS-server`*****`-priority-`*****`priority-level`***.
- When you create a resource management policy for priority levels and you use the **`source-file`** option to specify values, you must include the **`priority`** option on the command line.
- CPU sharing is not supported on Windows. As a result, when creating a resource management policy for priority levels on Windows, the **`cpu`** option is allowed, but will not be honored.
- The values that you assign to the **`global-casuser`**, **`global-casuser-hdfs`**, and **`local-tables`** attributes must be specified in bytes.
- Options that are specified on the command line take precedence over the options in the source file.

See Also

- "Create Policies from JSON Templates" in *SAS Viya Administration: Using the Command-Line Interfaces*
- "CAS Resource Management Policies" on page 102

Details

- When running the CAS CLI on a UNIX machine or a Macintosh machine, with the CAS server running on a Windows machine, a Windows pathname that contains backslashes must be enclosed in single quotation marks. Here is an example: **`'\\tmp\sas'`**.

- The CAS CLI supports the following environment variables:
  - SAS_CLI_DEFAULT_CAS_SERVER
  - SAS_CLI_DEFAULT_CAS_SESSION

  You can assign values to the environment variables that you want to remain in effect throughout your session. If the CAS CLI command requires the `server`, or `session-id` options, and the environment variables are set, then you can omit the required options from the CAS CLI command.

  For example, suppose the following:

  - **SAS_CLI_DEFAULT_CAS_SERVER** is set to `serverA`.

  You can then run this command without specifying the required **server** options: `./sas-admin cas admin-users list`.

  Note: Some commands do not support the use of some of the environment variables. For example, the CAS CLI ignores the CAS environment variables for the following command:

- sessions delete

  You must explicitly specify all required options when using this command.

See Also

- "Command-Line Interface: Overview" in *SAS Viya Administration: Using the Command-Line Interfaces*
- SAS Viya Administration: Identity Management
- SAS Viya Administration: Data

SAS Cloud Analytic Services: Concepts

CAS Controller

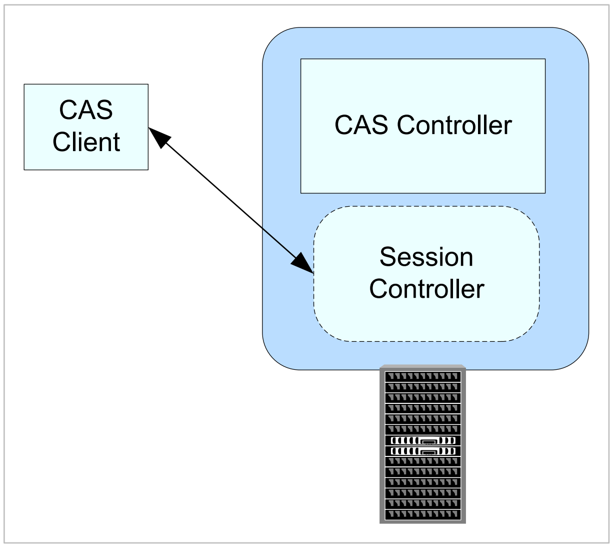
Controller is one of three roles that can be assigned to a machine for SAS Cloud Analytic Services (CAS): controller, backup controller, and worker. For both server architectures—distributed and single-machine—one machine is assigned the controller role. When the server starts, the controller process is started. This process is sometimes referred to as the server controller. The controller accepts connections from clients.

Single-machine CAS Server

The single-machine architecture uses symmetric multiprocessing (SMP). The functionality for a single-machine server is nearly identical to MPP, except that there is no cluster communication. In this architecture, the server acts as a controller. Before a client connects, the server listens on a port for connections.

After a client connects, a session is created and the session connects back to the client. (This is identical to the method that is performed by a CAS server that uses MPP. Compare with “Distributed CAS Server”.)

Figure 11 *Single-machine CAS Server*



CAS Backup Controller

A SAS Cloud Analytic Services (CAS) backup controller (sometimes referred to as *secondary controller*) provides fault tolerance for the CAS controller. A backup controller is used only in a distributed server architecture. Deploying a backup controller is optional. CAS supports one backup controller only.

Note: A requirement for operating CAS with a backup controller is that it and the CAS controller (the primary controller) must both use the same shared file system. For more information, see “Set Up a Shared File System for CAS Controllers (Post-Deployment)”.

When CAS starts, the backup controller process is also started. In the event that the controller experiences a disruption (such as a loss of network connectivity, disk full scenarios, and so on) the backup controller enables the CAS server to continue running. When the backup controller takes control of client communication, the transfer is seamless. For more information, see “Architecture” in *SAS Cloud Analytic Services: Fundamentals* .

CAS Workers

When a server is running in massively parallel processing (MPP) mode, in addition to a controller, the server also has multiple machines that are assigned the worker role.

The controller passes out work to each worker node. Each worker node sends the results of its computations back to the controller.

CAS Roles

Superusers have permission-exempt access to CAS (with the exception of access to data) and are exempt from all CAS authorization requirements.

In SAS Environment Manager, the Superuser role is never initially or automatically assumed. If you are a member of a CAS server's Superuser role, you can become a Superuser by explicitly assuming the role for that server. For example, you might assume the role to troubleshoot and resolve an access issue. After the issue is resolved, you relinquish the role.

The account that starts a CAS server is automatically assigned to that server's Superuser role.

Note: The following built-ins actions for SAS Cloud Analytic Services require a user ID that can assume the Superuser role:

- addNode
- installActionSet
- refreshLicense
- removeNode
- shutdown

For more information about the built-ins actions for SAS Cloud Analytic Services, see Builtins Action Set: Details.

Role	Description	Initial Members
Superuser	Provides permission-exempt access to a CAS server. Only a Superuser can perform the following tasks: ■ Stop the server. ■ Add and remove nodes. ■ Manage role membership. ■ See and manage the paths list. The account under which a CAS server runs is an implicit member of that server's Superuser role. Make sure that each CAS server has at least one other designated Superuser. **Note:** By default, the users that are assigned this role have permission-exempt access to metadata. However, they do not have permission-exempt access to data (CAS libraries). To give	SAS Administrators Process owner for the server Backup Administrator (sas.deploymentBackup) Report Distribution Service administrator account (sas.reportDistribution)

Role	Description	Initial Members
	users with this role permission-exempt access to data, you must modify access controls to explicitly grant them access.	Report Images Service administrator account (sas.reportImages) Scheduler Service administrator account (sas.scheduler) Report Alerts Service administrator account (sas.reportAlertsEval) VSD Service administrator account (sas.svi-vsd-service) Relationship Service (sas.relationships) Data Selection Service (sas.dataSelection)
Data	Assign members to this role only if you have users who should have permission-exempt access to data but should not be able to perform all administrative tasks. Not all interfaces support the Data role. **Note:** By default, the users that are assigned this role have permission-exempt access to metadata. However, they do not have permission-exempt access to data (CAS libraries). To give users with this role permission-exempt access to data, you must modify access controls to explicitly grant them access.	None
Action	Do not use this role. Not all interfaces support the Action role.	None

Note: The Data role provides a subset of the abilities of the Superuser role. You cannot be a member of both the Superuser role and the Data role in the same session.

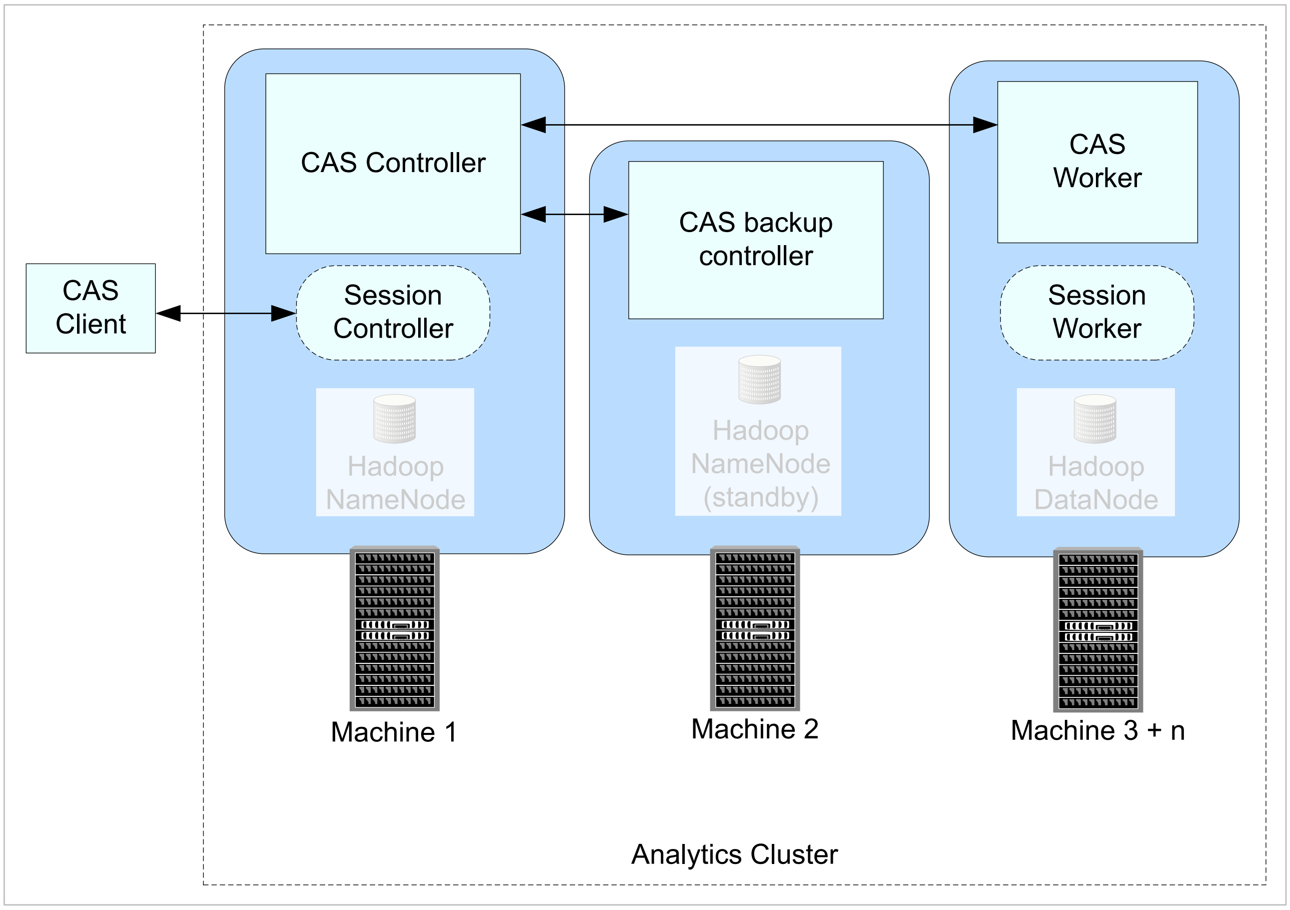
See Also

- Manage CAS Role Memberships in SAS Environment Manager on page 25
- Manage CAS Roles

Distributed CAS Server

CAS can be co-located with Hadoop on a cluster of machines. This massively parallel processing (MPP) architecture is appropriate for analyzing large data sets. Analysis proceeds on tables that are already made available to the server (loaded) or on tables that are gathered or created by the server on demand. A distributed CAS server consists of a controller, at least one worker, and a backup controller running on a minimum of two machines. (If a backup controller is deployed, then the minimum number of machines is three.)

Figure 12 *Distributed CAS Server*



Multiple CAS Servers

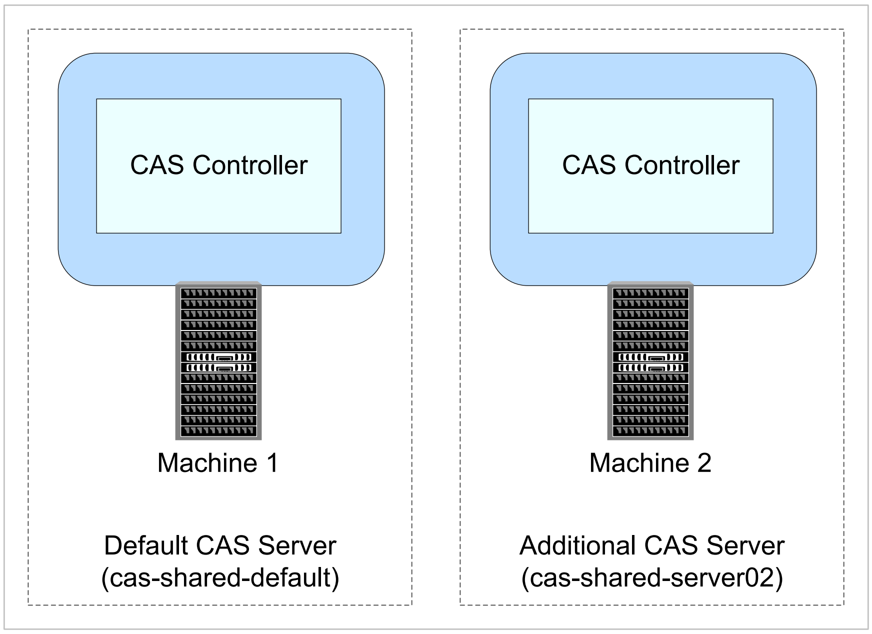
Overview

As of SAS Viya 3.4, it is now possible to have multiple instances of CAS servers within a single instance of SAS Viya.

What constitutes a CAS server depends on the type of CAS environment that you are running:

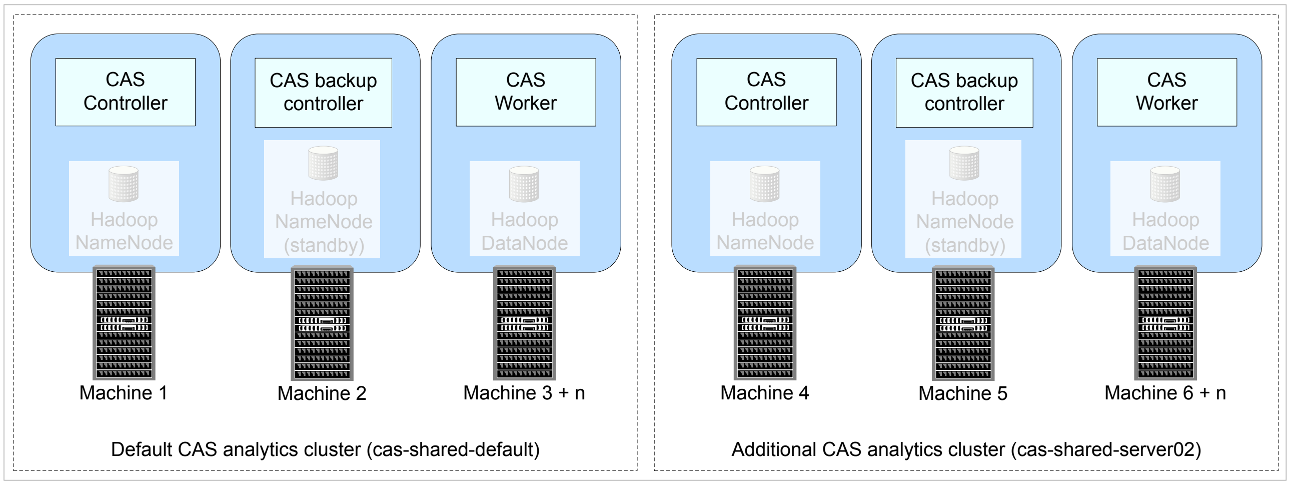
- In a symmetric multiprocessing (SMP) environment, a CAS server consists of a controller and runs on a single machine.

Figure 13 *Multiple Single-Machine CAS Servers (SMP Mode)*



- In a massively parallel processing (MPP) environment, a distributed CAS server consists of one controller, one or more workers, and one backup controller (optional) each running on a separate machine.

Figure 14 *Multiple Distributed CAS Servers (MPP Mode)*



Requirements

- You can add a CAS server to a machine that does not already host existing SAS Viya software.
- CAS servers that are added to a SAS Viya deployment cannot be removed, without removing your entire SAS Viya deployment.

- A multitenant environment does not support multiple CAS servers per tenant. (Each tenant has exclusive access to a single CAS Server.)
- In SAS Viya environments that have more than one CAS server, the default CAS server (typically, cas-shared-default) must be running. The default CAS server needs to be running even if a customer is using another CAS server, so that certain caslibs such as AppData, ReferenceData, and SystemData can be accessed from the default CAS server. (These caslibs contain data needed by applications such as SAS Visual Analytics, which depends on AppData for map data.) The default CAS server is defined in the `sas.casmanagement.global.casServer` configuration property.
- Make sure that you are licensed for the additional CAS servers that you are planning to add.

  When properly set, the CAS server option cas.MAXCORES specifies the limit for the total number of physical cores that are available to a CAS server. For more information, see cas.MAXCORES.
- Your SAS Viya deployment must be a Linux deployment.

  Multiple CAS servers are not supported on Windows environments.
- You are limited to one CAS controller or one CAS backup controller per machine.
- If you are adding a distributed CAS server that contains a backup controller, make sure that the backup controller and the CAS controller (the primary controller) both use the same shared file system.

  For more information, see "Set Up a Shared File System for CAS Controllers (Post-Deployment)" on page 15.
- Every machine on which you are installing a CAS server must contain the CAS user account (cas) and group (sas).

  For more information, see "Set Up the cas Account" in *SAS Viya for Linux: Deployment Guide*.
- For programming-only deployments and visual deployments that use the `CASHostAccountRequired` custom group, there is an additional requirement for users' home directories. In these two cases, a user's Casuser caslib is mapped to **`~/casuser`**. Therefore, the home directories (`$HOME`) for all CAS users must be shared so that they can be accessed from both the controller and the backup controller machines. Sharing users' home directories ensures that the path for the `CASUSER` library is available during CAS session start-up.

  For most other CAS session scenarios, the `CASUSER` library is set to a path in the shared file system described in "Set Up a Shared File System for CAS Controllers (Post-Deployment)".

See Also

"Add a CAS Server" on page 10

Session Processes

When a user connects to the server with a client, the server starts a session process for the user. Afterward, the client communicates with the session process.

A server running in symmetric multiprocessing mode (SMP mode) consists of a controller only, and the server starts a session controller process only. It is the session controller process that operates on rows of data.

In a distributed server (MPP mode), a session process is created on each machine in the cluster. These processes are sometimes referred to as the session controller and session worker processes.

Even though the sessions have their own operating system processes, the server processes must continue to run. When the server process terminates, the session processes also terminate.

Paths List

From a CAS server, all access to file system paths (host and HDFS directories) is through caslibs. To limit the paths that are available to non-administrators when they create or edit a caslib, use one of the following approaches:

- Create a denylist of paths that should not be available.
- Create an allowlist of paths that should be available.

You can view and modify the lists for CAS using:

- SAS Environment Manager.

  See, "Manage Path Lists (Allowlists and Denylists)".
- or, the programming interfaces.

  See, "Access Control Action Set" in *SAS Viya: System Programming Guide* .

Here are key points:

- Paths must be absolute.
- Paths must be unique.

  CAS automatically removes any duplicate paths.
- On Linux, if a denylist path is changed to a symbolic link, then the denylist should be updated using the fully resolved path.
- All subdirectories of each specified path are affected.
- Paths list constraints do not affect access to existing caslibs.
- If you do not define an allowlist or a denylist, no paths list constraints are in effect.
- Paths list constraints do not apply to users who assume the Superuser role or the Data role.
- Only users who assume the Superuser role for a server can see and manage that server's paths list.

Note: Access to third-party databases is not affected by a server's denylist or allowlist.

See Also

"Lock Down SAS Workspace Servers" in *SAS Viya Administration: Programming Run-Time Servers*

Caslib Management Privileges

Table 1 Caslib Management Privileges

Task	Who Can Perform the Task[1]
Add global caslibs.	Superusers and Data administrators. Users who have global caslib management privileges.
Add session caslibs.	Superusers and Data administrators. Users who have session caslib management privileges.
Delete global caslibs.	Superusers and Data administrators. Users who have global caslib management privileges can delete any global caslib for which they have the ReadInfo and ManageAccess permissions.
Delete session caslibs.	Superusers and Data administrators. Users who have session caslib management privileges can delete any session caslib for which they have the ReadInfo and ManageAccess permissions.
Adjust caslib management privileges.	Superusers and Data administrators.

1 Global caslib management privileges correspond to the ManageAccess permission on the _GLOBAL caslib. Session caslib management privileges correspond to the ManageAccess permission on the _SESSION caslib.

Note: Data administrators are displayed in CAS Server Monitor only.

See Also

- Adjust Caslib Management Privileges using SAS Environment Manager
- Manage Access to a Global Caslib using CAS Server Monitor

Playbooks for Adding Worker Nodes

Adding worker nodes to your CAS server is accomplished using the third-party orchestration tool, Ansible. When run on a CAS server that already has workers (MPP mode), or on a CAS server that does not have workers (SMP mode), Ansible performs these steps:

- configures SSH for the new machines and all existing machines that comprise the CAS server.

- installs software on all machines listed in the `sas_casserver_worker` group contained in the Ansible inventory file.

There are two playbooks to add nodes. Each playbook offers a different set of CAS usage characteristics:

- site playbook (site.yml)

  Designed for when you want to:

  - ☐ permanently add workers and have a maintenance window.

    Added workers are automatically joined to the CAS server.

  - ☐ change your CAS server configuration (for settings other than adding workers).

  Note: When you run site.yml playbook, Ansible stops and restarts the SAS Viya deployment.

- deploy-casworker playbook (deploy-casworker.yml)

  Designed for when you want to:

  - ☐

    permanently add workers, but do not have a maintenance window when the CAS server can be restarted.

    On first use, added worker nodes must be joined manually to the CAS server using SAS Environment Manager. New worker nodes added by running deploy-casworker.yml are automatically joined to the CAS server after any subsequent restarts of the CAS server.

    Adding a new worker node using SAS Environment Manager is necessary only when the CAS server is already running.

    For more information, see “Manage CAS Server Nodes” on page 24.

  - ☐ temporarily add workers that persist only until the CAS server restarts or until the worker is dropped manually using the SAS Environment Manager.

    Added workers must be joined manually to the CAS server using the CAS Environment Manager.

    For more information, see “Manage CAS Server Nodes” on page 24.

  Note: When you run deploy-casworker.yml playbook, Ansible does not restart the CAS server.

  TIP Instead of SAS Environment Manager, you can use CAS Server Monitor if it is enabled in your deployment.

  CAUTION
  Use the deploy-casworker playbook for adding worker nodes only. Do not change other CAS server configuration settings using the deploy-casworker playbook. Doing so can cause a mismatch between configuration in memory versus configuration on disk.

Understanding Configuration Files and Start-up Files

Several SAS applications require application-specific configuration and start-up before SAS Cloud Analytic Services begins processing client requests. It is worthwhile to understand how the files are processed so that you can use the same technique to customize your server deployment.

Configuration Home Directory

The installation and deployment software creates a configuration home directory for each server instance.

Here is an example:

```
/opt/sas/viya/config/etc/cas/default
```

The final directory in the path, *default*, is the deployment instance for the server.

Standard Configuration Files

The configuration home directory includes several files with standard names. The server automatically processes these files when the standard names are used.

The following table describes the purpose and use for each of the standard files.

Note: The files are listed in the order that they are processed at the server start up.

Table 2 Server Configuration Files

Standard File Name	Description
casconfig.lua	This file contains most of the configuration settings for the server instance, such as the network port that the server listens on. During deployment, RPM owns the casconfig.lua file and during updates can override any user configuration. For more information about the settings in the file, see “Configuration File Options” on page 64.
casconfig_deployment.lua	This file contains CAS configuration settings that are created during deployment by Ansible from vars.yml. During restarts and updates, user configuration settings are overwritten.
casconfig_usermods.lua	This file contains modifications made by the SAS administrator. Using casconfig_usermods.lua ensures that your modifications are not overwritten when you upgrade CAS.

Standard File Name	Description
conf.d/	This directory can contain one or more configuration files that are similar to the casconfig.lua file. The files are processed in alphabetical order. The files in this directory are processed before the casconfig.lua file.
node.lua	This file contains host-specific configuration. One possible use is for security setup that relies on the host name.
node_usermods.lua	This file contains modifications that are made by the SAS administrator. Using node_usermods.lua ensures that your modifications are not overwritten when you upgrade CAS.
logconfig.xml	This file contains the SAS logging facility instructions that control server logging.
perms.xml	This file contains the initial permission settings. This file is not used after the first time the server is started and the permstore is populated.
cas.hosts	This file contains the initial set of host names and roles (controller or worker) for the server.
cas.settings[1]	This file contains CAS and system environment variables that are created during deployment by Ansible from vars.yml. During updates, user configuration settings can be overwritten.
cas_usermods.settings	This file contains modifications that are made by the SAS administrator. Using cas_usermods.settings ensures that your modifications are not overwritten when you upgrade CAS.

1 There is a global version of cas.settings that resides in /opt/sas/viya/home/SASFoundation. CAS processes the global version of cas.settings **before** processing the configuration-specific version of cas.settings.

When the server starts, the configuration files that are described in the preceding table are processed. After the configuration is complete, the server runs start-up scripts.

The following table describes the standard names for the start-up files in the configuration home directory. The start-up scripts run before the server accepts any client connections. This is also referred to as *session-zero processing*.

Session zero runs under the CAS service account. Session zero already assumes the superuser role when the startup scripts are run. Make sure not to drop the role. Because session zero assumes the superuser role, you should be aware of the restrictions for data access for administrators. For more information, see “CAS Roles” on page 46.

Table 3 *Server Start-up Files (Session 0 Processing)*

Standard File Name	Description
casstartup.lua	This file contains the actions to run as the CAS server starts, such as some addFmtLib actions and a setServOpt action, that are created during deployment by Ansible from vars.yml. CAS processes casstartup.lua before any of the other start-up files residing in the `start.d/` directory.

Standard File Name	Description

casstartup_usermods.lua	This file contains modifications that are made by the SAS administrator to casstartup.lua, such as adding global-scope caslibs and loading global-scope tables. Using casstartup_usermods.lua ensures that your modifications are not overwritten when you upgrade CAS.
	
	CAS processes casstartup_usermods.lua before any of the other start-up files residing in the `start.d/` directory.
	
	Use Lua syntax such as the following. Do not forget to use global scope.
	
	`s:table_addCaslib{caslib="worldbank",`
	`     dataSource={srcType="path"},`
	`     path="/rdstore/data/smp/world_bank",`
	`     session=false}`

start.d/	This directory contains one or more start-up scripts that are similar to the casstartup.lua file. The files are processed in alphabetical order. The files in this directory are processed after the casstartup.lua file **IMPORTANT** CAS processes only files with a .lua file extension as start-up files.

Fault Tolerance

The distributed CAS server has a communication layer that supports fault tolerance. A distributed server can continue processing requests even after losing connectivity to some nodes. The communication layer also enables you to remove or add worker nodes from a server while it is running.

If a deployment uses a backup controller, the backup controller is started along with the rest of the nodes in the CAS server. The backup controller continuously tracks the state of the server, and is current if it has to take over control. The binary protocol is the socket communication between CAS client and corresponding server session that is typically provided using port 5570. Clients like SAS, Python, SWAT, and the CASClient Java library (a JAR) normally communicate using this interface.

If the primary controller fails, the site operates without fault tolerance for the controller until a planned outage. During the planned outage, the site should recover the failed controller and return to a redundant state.

After the outage, the primary controller accepts connections from clients and the backup (or secondary) controller resumes its role of providing fault tolerance.

If the backup controller fails or is shut down while the primary controller continues to operate, the site can continue to operate without fault tolerance.

Note: For information about fault tolerance in other parts of SAS Viya, see "Fault Tolerance in SAS Viya (Linux)" in *SAS Viya Administration: General Servers and Services*.

Operating system tuning such as ulimits should be identical on both controller hosts.

For sites that co-locate the server with Hadoop:

- You can set the primary controller and backup controller to use the same hosts as the active NameNodes and the standby NameNodes. This is not a requirement.
- The HADOOP_NAMENODE environment variable can include the two host names. You can specify the active and the standby NameNodes hosts, separated by a colon.

Access controls and caslib information are stored in a directory that is known as a permstore. While the primary controller and the backup controller are running, the permstore for the backup controller is kept in sync with the primary controller. After a failover, the permstore for the backup controller becomes the most current. Part of the task of restoring the primary controller is to copy the files from the permstore on the backup controller to the permstore on the primary controller. For more information, see *SAS Viya Administration: Backup and Restore*.

See Also

- "Set Up a Shared File System for CAS Controllers (Post-Deployment)"
- "Recover a Failed Controller"

CAS Resource Management

Overview

IMPORTANT CAS resource management use of CPU shares from SAS Configuration Server (Consul) applies only on Linux. But CAS resource management use of quotas is applicable on both Linux and Windows.

On Linux platforms, CAS has resource management capabilities that enable administrators to control CAS table size and CPU consumption. CAS relies on the Linux kernel feature, cgroups, to provide the CPU and memory consumption control. You must implement cgroups before you can fully use CAS resource management. For more information, see "(Optional) Additional Requirements for CAS Resource Management" in *SAS Viya for Linux: Deployment Guide*.

You can implement resource management broadly through the use of the cas.MEMORYSIZE and cas.CPUSHARES server configuration options, or more specifically through policies that you create with the SAS Viya command line interface. CAS resource management is achieved on the CAS server, with a policy or server option that applies to a specific server. If you have additional CAS servers on which you want to manage resources, then you must create a unique policy or server option definition for each. No steps need to be performed on the client side.

TIP CAS Java clients already handle all errors that are returned by the server—including those due to a lack of resources. However, Java clients can test specifically for CAS table quota failures by checking these two status code values: `SESSION_TABLE_QUOTA_EXCEEDED` and `GLOBAL_CASLIB_QUOTA_EXCEEDED`. Both of these Java constants reside in the `com.sas.actions.StatusCodes` class. Compare these constants to the value returned by `getStatusCode()` in `CASException`.

Policy Details

Here are some key details about CAS resource management policies:

- You must have CAS Superuser privileges to create and manage resource management policies.
- Supported only in full deployments.

> **IMPORTANT** For CAS to use resource management policies, the environment variable, CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT must be turned on. For more information, see CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT.

- CAS resource management use of CPU shares from SAS Configuration Server (Consul) applies only on Linux. But CAS resource management use of quotas is applicable on both Linux and Windows.
- Supported on both single-machine (SMP) and distributed (MPP) CAS servers.
- Policies are specific to a CAS server.
- Policy limits apply per machine for distributed CAS servers.
- Policies are managed through the SAS Viya command line interface and are turned off by default.

> **TIP** With the CAS command line interface, you can create a policy template that can serve as a starting point for creating your own CAS resource management policies. For more information, see "Create Policies from JSON Templates" in *SAS Viya Administration: Using the Command-Line Interfaces*.

- CAS stores policies as key-value pairs in the SAS Configuration Server (Consul).
- The SAS administrator can query the Consul to know the current settings and to monitor the policies.
- All the Consul settings are persisted across a maintenance update.

How Policies Work

Administrators create policies using the SAS Viya command line interface, which stores the policies as key-value pairs in the SAS Configuration Server (Consul).

There are two types of policies:

- global caslibs (`globalCaslibs`)

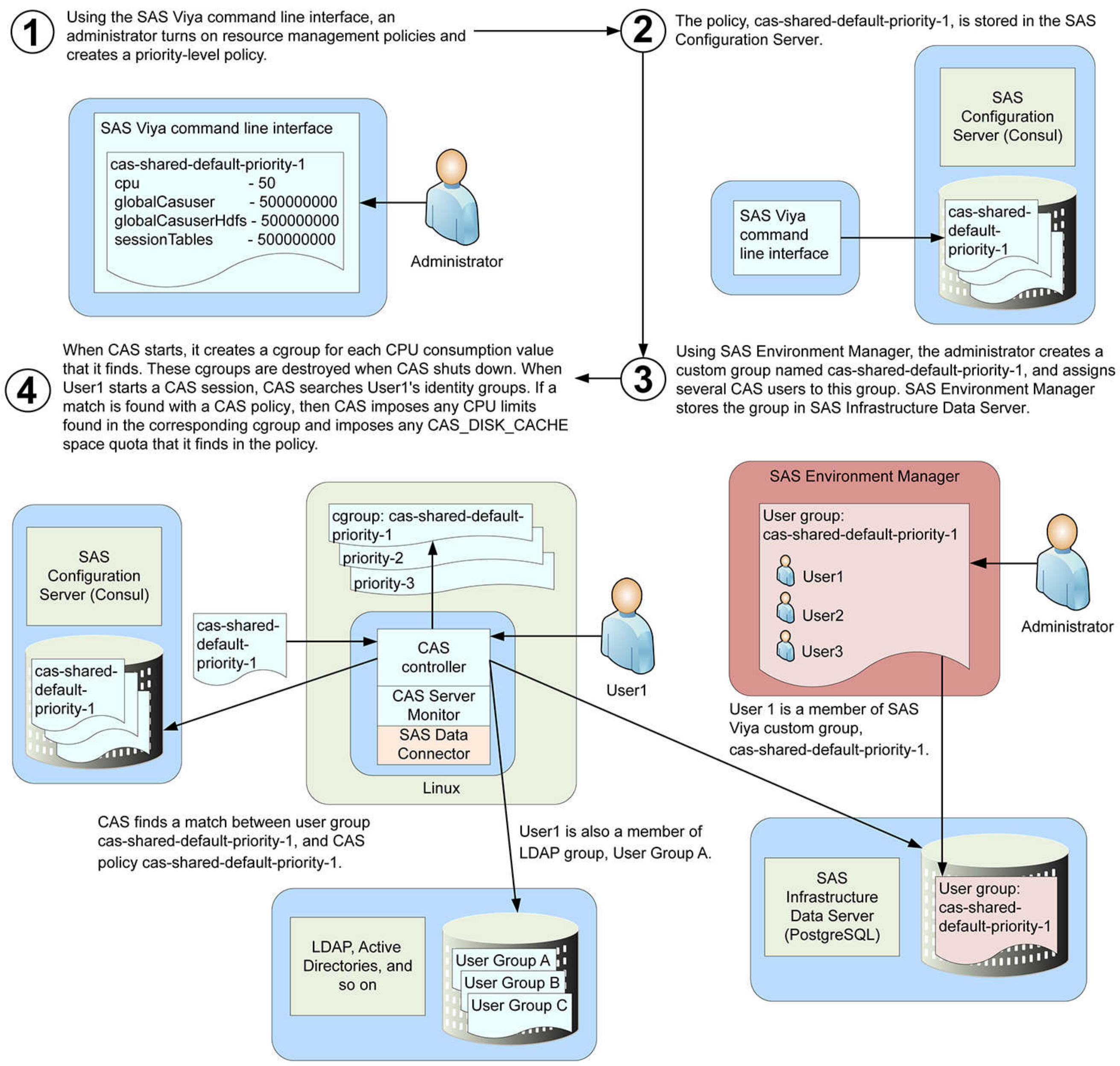
  One policy per CAS server that places a CAS_DISK_CACHE space quota on global caslibs.

- priority-level (`*CAS-server-name*-priority-*n*`)

  A maximum of five policies per CAS server that places a CAS_DISK_CACHE space quota on personal caslibs and a CPU consumption limit on CAS sessions.

If policies are defined, a CAS server reads its policy information directly from the configuration server. For each priority-level policy it finds that contains a CPU consumption value, CAS creates a corresponding Linux cgroup. These cgroups are created at CAS server start-up, and destroyed when the CAS server is shut down. CAS provides one policy for global caslibs and up to five priority level policies to which sites can assign resources. The administrator can assign users to these priority-level policies based on their identity group memberships, or they can explicitly assign priority-level policies to individual users.

Figure 15 How CAS Policies Work


1
Using the SAS Viya command line interface, an administrator turns on resource management policies and creates a priority-level policy.
2
The policy, cas-shared-default-priority-1, is stored in the SAS Configuration Server.
SAS Viya command line interface
cas-shared-default-priority-1
cpu - 50
globalCasuser - 500000000
globalCasuserHdfs - 500000000
sessionTables - 500000000
Administrator
SAS Configuration Server (Consul)
SAS Viya command line interface
cas-shared-default-priority-1
When CAS starts, it creates a cgroup for each CPU consumption value that it finds. These cgroups are destroyed when CAS shuts down. When User1 starts a CAS session, CAS searches User1's identity groups. If a match is found with a CAS policy, then CAS imposes any CPU limits found in the corresponding cgroup and imposes any CAS_DISK_CACHE space quota that it finds in the policy.
4
3
Using SAS Environment Manager, the administrator creates a custom group named cas-shared-default-priority-1, and assigns several CAS users to this group. SAS Environment Manager stores the group in SAS Infrastructure Data Server.
SAS Configuration Server (Consul)
cas-shared-default-priority-1
cgroup: cas-shared-default-priority-1
priority-2
priority-3
cas-shared-default-priority-1
CAS controller
CAS Server Monitor
SAS Data Connector
Linux
User1
SAS Environment Manager
User group: cas-shared-default-priority-1
User1
User2
User3
Administrator
User 1 is a member of SAS Viya custom group, cas-shared-default-priority-1.
CAS finds a match between user group cas-shared-default-priority-1, and CAS policy cas-shared-default-priority-1.
User1 is also a member of LDAP group, User Group A.
LDAP, Active Directories, and so on
User Group A
User Group B
User Group C
SAS Infrastructure Data Server (PostgreSQL)
User group: cas-shared-default-priority-1

To help to explain how policies work, here is a sample policy definition for a CAS server. In this definition, there is a policy for global caslibs, three priority-level policies (out of a possible of five), and priority-level policy assignments for three individual users:

```
resourceManagement
   globalCaslibs
      _ALL_       400000000
      HPS         200000000
      MyGlobal    100000000
```

```
priorityLevels
  cas-shared-default-priority-1
     cpu               - 50
     globalCasuser     - 500000000
     globalCasuserHdfs - 500000000
     sessionTables     - 500000000
  cas-shared-default-priority-2
     cpu               - 20
     globalCasuser     - 50000000
     globalCasuserHdfs - 50000000
     sessionTables    - 50000000
  cas-shared-default-priority-3
     cpu               - 30
     globalCasuser     - 10000000
     globalCasuserHdfs - 10000000
     sessionTables    - 10000000

priorityAssignments
   userA    1
   userB    3
   userC    2
```

CAS has built-in functionality to recognize user group names that start with the name of the CAS server as a group related to CAS resource management. When a user authenticates to create a new session, CAS always scans the user's identity groups searching for resource management user group names. If CAS finds that the user belongs to a resource management user group—a group whose name starts with the CAS server name—then CAS attempts to match the group name with the list of resource management policies.

Relying on this behavior, the administrator can create corresponding custom groups using identical names in SAS Environment Manager and add the appropriate user names to these groups.

In our example, if the user that is authenticating to CAS is a member of the group, **cas-shared-default-priority-2**, then CAS enforces the policy of the same name. Therefore, this user has a 50GB quota for all loaded tables and has a CPU share of 20. (CPU shares are discussed later in this document.)

If a user is a member of multiple resource management groups, then CAS assigns the lowest priority number. If the user is not a member of a resource management group, then CAS searches for an explicit assignment for the user under the policy definition `priorityAssignments` section. If the user is not a member of a resource management group, and the user has not been assigned any priority level, then the user has no limits.

CPU Shares (Linux)

IMPORTANT CAS resource management use of CPU shares from SAS Configuration Server (Consul) applies only on Linux. But CAS resource management use of quotas is applicable on both Linux and Windows.

On Linux, CAS servers support session CPU limits. Because there is a unique user for each CAS server session, CPU consumption can be controlled on a per user basis.

Note: You can specify cgroup shares for the CAS server as a whole with the cas.CPUSHARES server configuration option.

Every priority-level policy can be assigned a CPU share. A share is a value relative to the sum of all the shares used in a CAS server's policies. The recommended practice is to define shares for all policies to total 100—thus defining percentages.

Here is our example repeated again:

```
resourceManagement
   globalCaslibs
      _ALL_      400000000
      HPS        200000000
      MyGlobal   100000000

   priorityLevels
     cas-shared-default-priority-1
       cpu                - 50
       globalCasuser      - 500000000
       globalCasuserHdfs  - 500000000
       sessionTables     - 500000000
     cas-shared-default-priority-2
       cpu                - 20
       globalCasuser      - 50000000
       globalCasuserHdfs  - 50000000
       sessionTables     - 50000000
     cas-shared-default-priority-3
       cpu                - 30
       globalCasuser      - 10000000
       globalCasuserHdfs  - 10000000
       sessionTables     - 10000000

   priorityAssignments
      userA    1
      userB    3
      userC    2
```

In this example, when a session is created by a user who is a member of the `cas-shared-default-priority-2` user group (and is not a member of `cas-shared-default-priority-1`), then CAS selects the priority 2 resource management group for the session. And, the session is placed in the `cas-shared-default-priority-2` cgroup with a 20% share.

On a completely busy system, that user (together with all other users assigned to the priority 2 cgroup) are limited to using 20% of the CPU capacity. But, during off-hours, when there is excess CPU capacity, that same user might be able to consume all available capacity.

Note that priority 3 was assigned a higher percentage than priority 2. Does that mean it would be more advantageous to be assigned to priority 3? Probably not. Because there are likely many more users assigned to the priority 3 policy sharing 30% of the CPU capacity. When making share assignments for policies, be sure to consider the potential number of users and their load that might be members of the corresponding resource management user group.

See Also

"CAS Resource Management Policies"

Using CAS Server Monitor

What Is CAS Server Monitor?

CAS Server Monitor is a web application that you use to monitor your CAS Server and to perform some administration tasks.

IMPORTANT By default, CAS Server Monitor is turned off in full deployments of SAS Viya.

Monitor Cheat Sheet

In addition to providing information, CAS Server Monitor supports the following tasks:

Task	Required Role
Stop the server	CAS (Superuser)
Add or remove nodes[2]	CAS (Superuser)
Terminate your sessions	(none)
Terminate other sessions	CAS (Superuser) or Data
Designate administrators	CAS (Superuser)
Set caslib permissions	CAS (Superuser) or Data[1]

1 In addition, any user who has the ManageAccess permission for a global caslib can set permissions on that caslib.

2 On Cloud Foundry, do not attempt to add nodes, remove nodes, or terminate a server instance from the CAS Server Monitor (or with the addNode and removeNode CAS actions). Instead, use the appropriate BOSH command.

Access the Monitor

IMPORTANT By default, CAS Server Monitor is turned off in full deployments of SAS Viya. Instead, use SAS Environment Manager.

1 You can access the monitor without first starting a CAS server session. However, if there are no sessions, the session list in the monitor is empty.

   If you already have a CAS server session, skip to Step 2. Otherwise, to start a session, perform the following steps:

   a Open a web browser and sign in to SAS Studio with administrator privileges:

   https://*reverse-proxy-server*/SASStudio

   b In the **Code** tab, start a CAS server session by entering the following:

   ```
   cas my_session;
   ```

   c Click 🏃.

   You should see output similar to the following:

```
56         cas my_session;
 NOTE: The session MY_SESSION connected successfully to Cloud Analytic Services 10.120.9.159 using
port 5570. The UUID is
       5120eb8f-ca06-8c44-9f93-2dd2e557b1cb. The user is myacct and the active caslib is
CASUSER(myacct).
 NOTE: The SAS option SESSREF was updated with the value MY_SESSION.
 NOTE: The SAS macro _SESSREF_ was updated with the value MY_SESSION.
 NOTE: The session is using 0 workers.
```

2 Open a web browser and enter the following URL in the address field:

   https://*http-reverse-proxy-machine-name*/cas-shared-*deployment-instance-name*-http

   Here is an example:

   https://myproxy.example.com/cas-shared-default-http

   TIP To designate administrators, select **Configuration** ⇨ **Administrators**, click **Add**, and select **CAS** (Superuser).

Usage notes:

- When the CAS server terminates, the CAS Server Monitor also terminates.
- The session view remains displayed even if the session is terminated. A session view is displayed until you close it.
- Sessions are removed from the list if the session is terminated. You can click the refresh button to get the current list of sessions.
- You can also access CAS Server Monitor from SAS Studio.
- CAS Server Monitor does not use a session time-out. You must click ⌄ in the top right corner of the window and select **Sign Out** to exit.

Set Monitor Preferences

To control the view that you see by default, click ⌄ in the top right corner of the window and select **Settings**.

SAS Cloud Analytic Services: Reference

Configuration File Options

How Do I Use CAS Configuration File Options?

You set SAS Cloud Analytic Services options in the CAS controller's configuration file, casconfig_usermods.lua. During CAS server start-up, the controller shares the configuration as each worker and the backup controller (if present) connects. By default, casconfig_usermods.lua is located in `/opt/sas/viya/config/etc/cas/default` on Linux and in `\ProgramData\SAS\Viya\etc\cas\default` on Windows.

If you want to isolate a configuration option change to a particular CAS node, then make your change in node_usermods.lua residing on the particular node machine.

TIP For sites that use Ansible, it is recommended that you make your CAS server configuration changes to vars.yml and rerun Ansible to apply these changes. For more information, see "Modify the vars.yml File" in *SAS Viya for Linux: Deployment Guide*.

There are additional CAS configuration files and directories. For more information, see "Understanding Configuration Files and Start-up Files".

For the order of precedence for server configuration options, see How the Session Option Values Are Determined.

When a session starts, session options specified in the casconfig files are set for the session. For the order of precedence for session options specified in the casconfig files, see Table 2 on page 54.

TIP Remember that you can also set operating system environment variables in casconfig_usermods.lua. Here is an example, `env.HADOOP_HOME='/hadoop/hadooop-someversion' env.HADOOP_NAMENODE='name_node.example.com'`.

You can override CAS configuration file settings for a session by changing the equivalent session option. For more information, see Setting Session Options.

Configuration File Options Reference

Note:

Other security-related configuration file options can be found in "Configuration File Options for Data Transfer" in *Encryption in SAS Viya: Data in Motion*.

cas.APPTAG='*tag-string*'

specifies an arbitrary string to prefix to log messages.

Using `apptag` helps determine which log messages are associated with an application.

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Session
Default	No *tag-string*
Note	The CAS server uses `apptag` when writing to its log.
See	cas.LOGCFGLOC
Example	`apptag='my_app'`

cas.AZUREAUTHCACHELOC="*path*"

specifies a location that is used to cache login information for connecting to an Azure storage system. Specifying this setting enables device code authentication in Microsoft Azure.

Valid in	casconfig_usermods.lua file
Category	Data

cas.AZURETENANTID="*string*"

specifies the tenant ID for connecting to an Azure storage system. The maximum length of the value is 64 bytes. Only alphanumeric characters and the hyphen character (-) are supported.

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Data

cas.CMPOPT='*optimization-value <optimization-value* <…>>' | 'all' | 'none'

specifies the type of code generation optimizations to use in the SAS language compiler.

- *optimization-value*

  specifies the type of optimization that the SAS compiler is to use. Specify one or more of the following as a space-delimited list enclosed in quotation marks:

  - 'dumptkgcode' | 'nodumptkgcode'

    specifies whether all CAS server nodes create an output file with the generated program. The CAS log lists where CAS writes the output file.

  - 'extramath' | 'noextramath'

    specifies whether the compiler is to retain or remove the extra mathematical operations that do not affect the outcome of a statement.

  - 'funcdifferencing' | 'nofuncdifferencing'

    specify **funcdifferencing** to calculate numeric-differencing derivatives for user-defined functions. Specify **nofuncdifferencing** to calculate analytic derivatives for user-defined functions.

  - 'guardcheck' | 'noguardcheck'

    specifies whether the compiler checks for array boundary problems.

Note: **noguardcheck** is set when cmpopt is set to 'all' or 'none'.

- □ 'misscheck' | 'nomisscheck'

  specifies whether to check for missing values in the data.

- □ 'precise' | 'noprecise'

  specify precise to handle exceptions at the operation boundary. Specify **noprecise** to handle exceptions at the statement boundary.

- ■ 'all'

  specifies that the compiler is to optimize the machine language code by using the **noextramath**, **nomisscheck**, **noprecise**, **noguardcheck**, and **nofuncdifferencing** optimization values.

  Note: 'all' cannot be specified with other values.

- ■ 'none'

  specifies that the compiler is not set to optimize the machine language code by using the **extramath**, **misscheck**, **precise**, **noguardcheck**, and **funcdifferencing** optimization values.

  Note: 'none' cannot be specified with other values.

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	Action
Default	**noextramath**, **nofuncdifferencing**, **noguardcheck**, **nomisscheck**, and **noprecise**
Note	If the data contains a significant amount of missing data, specify **misscheck** to optimize the compilation. Otherwise, specify **nomisscheck**.
Example	In this example, the SAS compiler is set to retain the extra mathematical operations, check for missing values, and handle exceptions at an operation boundary: `cas.cmpopt='extramath misscheck precise'`

cas.COLLATE='mva' | 'uca'

specifies the collating sequence for sorting.

`mva` specifies SAS client collating. `uca` specifies a locale-appropriate collating sequence.

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	Sort
Default	**'uca'**
Example	`cas.collate='mva'`

cas.COLOCATION='none' | 'hdfs'

specifies whether to create a personal caslib (`hdfs`) at CAS server start-up.

A server started in MPP mode defaults to `hdfs` because it assumes it is co-located with Hadoop. Specify `none` for the server running in MPP mode not to create a personal caslib at start-up.

Valid in	casconfig_usermods.lua file
Category	Caslib
Default	**`cas.colocation='hdfs'`**
Restriction	Applies to Linux only.
Requirement	Used with `cas.mode='mpp'` and `cas.hdfsuserloc`.
Example	In this example, the CAS server is running in MPP mode and is not co-located with Hadoop. At start-up, the CAS server does not create a personal caslib for the user ID under which the server is run. `cas.colocation='none'`

cas.CPUSHARES='*number*'

on Linux operating systems, specifies cgroup shares for the CAS server as a whole. The higher the value, the more priority is given to the CAS server when Linux allocates CPU resources.

on Windows operating systems, specifies the percentage of CPU time that should be allocated to the CAS server. The range on Windows is 0–100. The default is 95. A zero indicates that the CAS server should choose a value between 1–100 that is appropriate for most environments.

Note: A recommended value for `cas.CPUSHARES` for a SAS Viya full deployment on Windows is 70.

Valid in	casconfig_usermods.lua file
Category	Administration
Linux specifics	Use cas.CPUSHARES to balance a CAS server against other processes in other top-level cgroups, including other CAS servers.
Windows specifics	The exact division of CPU time between CAS and the other processes is performed by the Windows operating system. CAS provides guidance only to Windows as to how this division should be performed. A value above 95 is not recommended. The maximum value (100) results in CAS not requesting that the operating system to limit the CPU cycles provided to the CAS server. A value of 100 can result in other critical processes, possibly including other processes that are essential parts of a SAS Viya installation, becoming unresponsive, and potentially failing.
Note	When other processes are idle, CAS can still consume up to 100% of available CPU cycles. CAS resource management use of CPU shares from SAS Configuration Server (Consul) applies only on Linux. But, CAS resource management use of quotas is applicable on both Linux and Windows.
See	CPU Shares (Linux)
Example	In this example, on Windows, at least 10% of the available CPU time is reserved for other processes, with up to 90% of the available CPU time being allocated to CAS processes: `cas.cpushares='90'`

cas.DATASTEPFMTERR=true | false

corresponds to the FMTERR in SAS. Specifies how the DATA step reacts when a format is not available. When `true`, the DATA step writes an error and stops. When `false`, the DATA step uses `$w.` or `BEST12.` instead of the unavailable format. (The unavailable format is still associated with variables in the output table.)

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	DATA Step
Alias	FMTERR
Default	True
Note	The values `true` and `false` are case sensitive.
See	FMTERR System Option
Example	In this example, the DATA step uses $w. or BEST12. instead of the unavailable format. `cas.datastepfmterr=false`

cas.DATASTEPMSGSUMLEVEL='all' | 'none' | 'put'

specifies the DATA step message summary level. When the DATA step runs on multiple threads, the same message can be generated on each thread. This option controls the summary level of duplicate messages.

- 'all'

  The first occurrence of all message and put statements are sent to the client when they occur. Duplicate occurrences of all message and put statements are summarized and sent to the client when the DATA step exits. This is the default.

- 'none'

  All message and put statements from every thread are written to the client log. No summarization occurs.

- 'put'

  The first occurrence of all message and put statements are sent to the client. Duplicate occurrences of messages are summarized and sent to the client when the DATA step exits. Put statements are not summarized; rather, they are sent to the client when they occur.

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	DATA Step
Default	All
Example	In this example, all message and put statements from every thread are written to the client log. No summarization occurs. `cas.datastepmsgsumlevel='none'`

cas.DATASTEPREPLACETABLE=true | false

specifies whether a DATA step can replace an existing table.

Valid in	CAS statement SESSOPTS option

	casconfig_usermods.lua file
Category	DATA Step
Default	true
Note	The values `true` and `false` are case sensitive.
Example	`cas.datastepreplacetable=true`

cas.DATASTEPMERGENOBY=‘ERROR’ | ‘NOWARN’ | ‘WARN’

specifies the dataStepMergeNoBy parameter. When a merge is run with no BY statement, this parameter determines whether no warning, a warning, or an error message is generated.

- ERROR

  specifies that, when there is no BY statement, the merge does not run, and an error message is generated.
- NOWARN

  specifies that, when there is no BY statement, the merge runs with no warning messages.
- WARN

  specifies that, when there is no BY statement, the merge runs with warning messages.

Valid in	CAS statement SESSOPTS option
Category	DATA Step
Default	NOWARN
Example	cas.dataStepMergeNoBy=‘WARN’

cas.DCHOSTNAMERESOLUTION= '[ep | ep_fqdn | cas | cas_ipv6]'

specifies how CAS sends the CAS node machine name to SAS Embedded Process. cas.DCHOSTNAMERESOLUTION is valid for any data store that supports SAS Embedded Process.

- 'ep'

  Send CAS node machine names to SAS Embedded Process exactly how they are defined in cas.hosts. SAS Embedded Process resolves the names using either IPv4 or IPv6.
- 'ep_fqdn'

  Send CAS node machine names to SAS Embedded Process as fully qualified domain names. SAS Embedded Process resolves the names.
- 'cas'

  (Default) send CAS node machine names to SAS Embedded Process as IPv4 addresses.
- 'cas_ipv6'

  Send CAS node machine names to SAS Embedded Process as either IPv4 or IPv6 addresses.

Valid in	casconfig_usermods.lua file
Category	Network
Default	'cas'
Restriction	Applies to Linux only.

Example	`cas.dchostnameresolution='ep'`

cas.DCPORT=*port number*

specifies the port number that CAS can use to listen for connections from an external data provider.

A value of -1 indicates that data connectors should share the port specified for GCPORT. A value of 0 indicates that data connectors should use an ephemeral port.

Valid in	casconfig_usermods.lua file
Category	Network
Default	-1
Range	-1 - 65535
Restriction	Applies to Linux only.

cas.DCEXTPORT=*port number*

specifies the port number that an external data provider, such as SAS Embedded Process, can use to connect to this node.

A value of 0 indicates that the data provider should use the actual port number that CAS is using to listen for connections. A non-zero value might be required in some cloud-based configurations.

Valid in	casconfig_usermods.lua file
Category	Network
Default	0
Range	0 - 65535
Restriction	Applies to Linux only.

cas.EXTHOSTKNOWNBY=‘*hostname*’

specifies the host name that external data providers should use to connect to this node.

This value might need to be set in some cloud configurations.

Valid in	casconfig_usermods.lua file
Category	Network
Restriction	Applies to Linux only.

cas.DQLOCALE='*locale-code*'

specifies the default locale to use for data quality (DQ) operations, using the five-letter SAS Quality Knowledge Base (QKB) ISO locale code.

For more information, see QKB Locale ISO Codes.

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	Data Quality
Example	In this example, the default locale for DQ operations is French Canadian: `cas.dqlocale='fr_CA'`

cas.DQSETUPLOC='*QKB-name*'

specifies the name of the default SAS Quality Knowledge Base (QKB) to use for data quality (DQ) operations.

QKB-name is the absolute path to a SAS Quality Knowledge Base.

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Data Quality
Example	`dqSetupLoc='/opt/sashome/SASQualityKnowledgeBases/en/my_qkb'`

cas.ELASTIC=true | false

indicates that new machines are allowed to join the analytics cluster.

Valid in	casconfig_usermods.lua file
Category	Administration
Default	false
Restriction	Applies to Linux only.
Requirement	Used with `cas.gcport`.
Supports	CAS servers running MPP.
Note	The values `true` and `false` are case sensitive.
See	cas.GCPORT
Example	In this example, the CAS controller allows new worker nodes to join the analytic cluster: `cas.elastic=true`

cas.EVENTDS='*event-data-set*'

specifies one or more event objects that define custom date events.

event-data-set specifies the name of a data set that contains event definitions. You can use a one-level name or a two-level name, such as `libref.dataset`. When specifying multiple names, separate each name with a space.

Enclose *event-data-set* in single quotation marks.

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Input Control
Example	`cas.eventds='mydataset'`

cas.FMTSEARCH='*logicalformatlibname logicalformatlibname2 …* '

specifies the format search list to be automatically set at session start-up.

Server configuration files can be used to add and promote common format libraries when a server starts. Promotion makes the format libraries available to all sessions.

During session start-up, the cas.FMTSEARCH value is used to generate and execute the setFmtsearch action. The setFmtsearch action specifies the order in which format libraries are searched. Table variables can have a user-defined format association. During table processing, when a user-defined format needs to be applied, CAS searches the list of format libraries established by the setFmtsearch action.

Category	Formats
Default	blank
Interaction	Specifying the setServOpt action for format search in startup.lua takes precedence over cas.FMTSEARCH used in casconfig.lua.
Note	The format library names are logical names known to a session. The names are case insensitive.
Example	`cas.fmtsearch='myformatsA myformatsB'`

cas.GCPORT=*port*

specifies the network port that is used on a distributed server for communication between the controller and its worker nodes.

The commonly configured port is 5580.

Valid in	casconfig_usermods.lua file
Category	Network
Default	0 (random port in the range 32678–61000)
Restriction	Applies to Linux only.
Supports	CAS servers running MPP.
See	cas.HTTPPORT and cas.PORT
Example	`cas.gcport=5580`

cas.HDFSUSERLOC='/*hdfs-path*/%USER'

for CAS servers running in MPP mode, specifies that the server create a personal caslib for each user at session start-up time in the specified HDFS path.

`'`*`hdfs-path`*`/%USER'` refers to a directory named for the user's user ID under the specified HDFS path. Make sure that %USER is always placed at the end of the path that is specified for the option value.

Enclose *hdfs-path* in single quotation marks.

Valid in	casconfig_usermods.lua file
Category	Data
Restriction	Applies to Linux only.
Requirement	cas.MODE='mpp' on page 80 and cas.COLOCATION='hdfs' on page 66
Example	In this example, the user's caslib directory is a subdirectory named for the user ID under **`/user`**: `cas.hdfsuserloc='/user/%USER'`

cas.HOSTKNOWNBY='*machine-name* | *IP-address*'

specifies the preferred network interface for other machines in the CAS grid to use to connect to this machine. This variable might be useful on machines that contain multiple network interface cards.

machine-name is the fully qualified host name that is associated with the preferred network interface. *IP-address* is the IP address of the preferred network interface.

On the controller machine, the host names in the machine list file should be either the IP addresses of the preferred network interface to use on each peer node or the fully qualified host name that is associated with the preferred network interfaces.

When `cas.ELASTIC=true` every machine in the CAS analytic cluster should specify `cas.HOSTKNOWNBY` and the `cas.ELASTIC=true` options.

Valid in	casconfig_usermods.lua file
Category	Network
Restriction	Applies to Linux only.
See	cas.ELASTIC and cas.MACHINELIST
Example	`cas.hostknownby='primary.private.example.com'`

cas.HTTPPORT=*port* | *port-range*

The port (or range of ports) that SAS Cloud Analytic Services listens to for HTTP communication.

The commonly configured port is 8777.

Valid in	casconfig_usermods.lua file
Category	Network
Default	0 (random port)
Note	If the first port in the range is already taken, CAS tries the next port until it finds a port that is free.
See	cas.HTTPPORTMAX , cas.GCPORT , and cas.PORT
Examples	`cas.httpport=8777`
	`cas.httpport=8777-9000`

cas.HTTPPORTMAX=*maximum-port-range*

specifies the maximum port range that SAS Cloud Analytic Services listens to for HTTP communication.

Valid in	casconfig_usermods.lua file
Category	Network
Default	0
Range	0–65535
See	cas.HTTPPORT
Example	`cas.httpport=8777-9000`

cas.INITIALBACKUPS= -1 | 0 | *positive-number*

specifies whether SAS Cloud Analytic Services (CAS) waits for backup controllers to connect to the CAS analytics cluster before CAS begins to accept client connections.

Valid values are:

- -1

  Use the value specified in cas.hosts for the number of backup controllers to connect to the analytics cluster before CAS begins accepting connections from clients.

- 0 (zero)

  Do not wait for any backup controllers to connect to the analytics cluster before CAS begins accepting connections from clients.

- 1

  Wait for the backup controller to connect to the analytics cluster before CAS begins accepting connections from clients.

Valid in	casconfig_usermods.lua file
Category	Server
Default	-1
Range	-1–1
Restriction	Applies to Linux only.
Example	`cas.INITIALBACKUPS=-1`

cas.INITIALWORKERS='*n*'

specifies the number of CAS worker nodes that must join the analytic cluster before CAS begins processing user connections.

cas.initialworkers enables administrators to establish an expected cluster size for configurations, where it is typical for all or most worker nodes to join elastically.

Valid in	casconfig_usermods.lua file
Category	Server
Default	-1
Range	-1 to 32767
Restriction	Applies to Linux only.
Requirement	cas.elastic must be set to true.
Notes	A value of zero indicates that the controller does not wait for any worker nodes to join the cluster before it begins to establish user connections. A value of -1 indicates that the controller waits for the number of workers that is specified in the machine list file.
See	cas.ELASTIC and cas.MACHINELIST
Example	In this example, the CAS controller waits for 16 workers to join the analytic cluster before it begins processing user connections. `cas.initialworkers='16'`

cas.INTERVALDS='*interval-1=libref.dataset-name-1 <interval-2=libref.dataset-name-2 …>*'

specifies one or more interval-name=value pairs, where the value is the name of a data set that contains user-defined intervals.

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	Input Control
See	INTERVALDS= System Option
Example	`cas.intervalds='subsid1=subsid.storeHours'`

cas.JREOPTIONS='(*JRE-option <JRE-option> <…>>*)'

specifies the Java Virtual Machine (JVM) options that SAS Cloud Analytic Services uses at start-up. Separate JRE options with a whitespace character. Enclose any paths in quotation marks.

For the list of the valid Java options, and what they do, see http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html

Valid in	casconfig_usermods.lua file
Category	Java
Default	(null)
Example	In the following example, the initial and maximum sizes of the memory allocation pool are set to 256 anc 1024MB, respectively. Also, the log4j configuration file path and Java classpath are set:

```
cas.jreoptions = '(-Xms256m -Xmx1024m
  -Dlog4j.configuration=' .. ' -Djava.class.path=' .. env.CAS_HOME .. '/lib/base/base-tkjni.jar')'
```

cas.KEYFILE='*pathname*'

identifies to the CAS controller the path and file name to the X.509 digital certificate file that is used to start the server. The certificate must be signed by a CA that is trusted by the CAS server.

Enclose *pathname* in single quotation marks.

Valid in	casconfig_usermods.lua file
Category	Security
Restriction	Applies to Linux only.
Requirement	Used with `cas.elasticssl` and `cas.mode='mpp'`.
Supports	CAS servers running MPP.
See	cas.GCPORT and cas.ELASTIC
Example	`cas.keyfile='/opt/TKGrid/certs/controller.pem'`

cas.LIFETIME=*minutes*

indicates the duration, in minutes, that a server remains running.

Valid in	casconfig_usermods.lua file
Category	Administration
Default	0

Example	In the following example, the server shuts itself down in 120 minutes: `cas.lifetime=120`

cas.LOCALE='*POSIX-locale-string*'

specifies the locale to use for sorting and formatting. For a list of valid POSIX locale strings, see SAS National Language Support (NLS): Reference Guide

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Localization
Default	'en_US'
Example	`cas.locale='fr_FR'`

cas.LOGCFGLOC='*pathname*'

specifies the path to the SAS logging facility logging configuration file.

Enclose *pathname* in single quotation marks.

Valid in	casconfig_usermods.lua file
Category	Log
See	cas.APPTAG on page 65
Examples	`cas.logcfgloc='/opt/sas/cas1/etc/logconfig.xml'`
	Here is an example on Windows: `cas.logcfgloc='C:\\ProgramData\\SAS\\Viya\\etc\\cas\\default\\logconfig.xml'`

cas.LOGFLUSHTIME=-1 | 0 | *number*

specifies the log flush time, in milliseconds.

- -1

  flushes logs after each action completes.
- 0

  flushes logs as they are produced.
- *number*

  flushes logs in *number* milliseconds.

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Log
Default	100
Range	-1–86400
Example	In the following example, the CAS server writes buffered lines to the log every 500 milliseconds: `cas.logflushtime=500`

cas.MACHINELIST='*path*/*machine-list-file*'

identifies the path and file name on the controller machine that contains the list of machines in the CAS analytics cluster.

Enclose *path* in single quotation marks.

machine-list-file contains all of the machines in the analytics cluster in the form:

`<`*`fully-qualified-domain-name`*` controller | worker>`

Place each machine on a separate line. For example:

```
my_machine01.example.com controller
my_machine02.example.com worker
my_machine03.example.com worker
my_machine04.example.com worker
my_machine05.example.com worker
```

Valid in	casconfig_usermods.lua file
Category	Administration
Restriction	Applies to Linux only.
Requirement	Used with `cas.mode='mpp'`.
Interaction	Do not specify `cas.mode='smp'` when a valid machine list file is used.
See	cas.MODE
Example	`cas.machinelist='/etc/my_machine_list'`

cas.MAXCORES='*number-of-cores*'

specifies the limit for the total number of physical cores that are available to a CAS server. For distributed CAS servers, cas.MAXCORES refers to the total number of cores for the controller and workers. (Cores for the backup controller are not counted.)

When specified, the lesser of the specified `cas.MAXCORES` value and the product licensed cores is used during CAS action invocation.

If hyperthreading is enabled on a worker, CAS uses two virtual cores, both on the same physical core. The controller also allocates threads. (See the examples that follow for more information.)

Valid in	casconfig_usermods.lua file
Category	Administration
Interaction	If cas.MAXCORES is specified too low, CAS is not able to establish a session (TKCASA_LICENSE_TOO_SMALL).
Note	The core count limit is server-wide, and for distributed CAS servers the value should be at least the same as the total number of machines. The total number of machines is the primary controller and workers. (The backup controller is not included in this total.) For example, if a distributed CAS server has one controller and one worker, and `cas.MAXCORES=4`, the maximum number of cores that the worker can use is two. If you set `cas.MAXCORES` too low, CAS writes a licensing error.
Examples	In this example, the total number of cores available to the CAS server is 17:

```
cas.maxcores='17'
```

In another example, we want to ensure that exactly 128 hyperthreads per worker are run. (Hyperthreads equal two times the number of cores.) For a single-machine CAS server:

```
cas.MAXCORES='64'
```

For a distributed CAS server use the formula, (Nworkers+1) * 64. To ensure that 128 hyperthreads per worker are run for a distributed CAS server that has a controller plus eight workers:

```
cas.MAXCORES='576'
```

cas.MAXSESSIONS='*n*'

specifies the maximum number of concurrent sessions. Users who can assume an administrative role are not subject to the limit.

Valid in	casconfig_usermods.lua file
Category	Server
Default	5000
Range	0–100000
Notes	Specifying zero (0) indicates that there is no session limit. This option cannot be changed after the system initializes.
Example	In this example, the maximum number of concurrent CAS sessions is 1,000: `cas.maxsessions='1000'`

cas.MAXTABLEMEM=*number* | '[*number* k | m | g | t] '

specifies the maximum amount of physical memory to allocate for a table.

TIP

The intent of cas.MAXTABLEMEM is to manage the efficiency of accessing CAS tables on disk, not to control the amount of data that CAS keeps resident in RAM. When you need to manage CAS memory, consider modifying cas.MEMORYSIZE.

- *number*

  specifies the maximum amount of physical memory, in bytes, to allocate for a table.

- '[*number* k | m | g | t]'

  specifies the maximum amount of physical memory to allocate for a table in a unit other than bytes: **k** (kilobytes), **m** (megabytes), **g** (gigabytes), and **t** (terabytes).

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	Caslib
Default	16M
Note	After this threshold is reached, the server uses temporary files and operating system facilities for memory management.

See cas.MEMORYSIZE

Example In this example, the CAS server can allocate up to 32MB of physical memory for a table:

```
cas.maxtablemem='32m'
```

cas.MEMORYSIZE=*number* | '[*number* k | m | g | t] '

specifies the maximum amount of physical memory to allocate for the CAS cgroup for each machine. This limit also applies to the YARN request, when cas.USEYARN is specified.

- *number*

  specifies the maximum amount of physical memory, in bytes, to allocate for the CAS cgroup and the YARN request.

- '[*number* k | m | g | t]'

  specifies the maximum amount of physical memory to allocate for the CAS CGroup and the YARN request in a unit other than bytes: **k** (kilobytes), **m** (megabytes), **g** (gigabytes), and **t** (terabytes).

Valid in casconfig_usermods.lua file

Category Administration

Default 0

Restriction Applies to Linux only.

See cas.USEYARN on page 86

“How to Limit Memory Use” in *SAS Cloud Analytic Services: Fundamentals*

Example In the following example, the maximum amount of physical memory allocated for the CAS cgroup and the YARN request is 256GB:

```
cas.memorysize='256g'
```

cas.MESSAGELEVEL='all' | 'default' | 'error' | 'none' | 'note' | 'warning'

specifies the log message level.

Valid in CAS statement SESSOPTS option

casconfig_usermods.lua file

Category Log

Default 'all'

Example `cas.messagelevel='default'`

cas.METRICS=true | false

causes CAS server metrics information to be displayed (`true`) or not displayed (`false`) in the SAS log.

When `cas.metrics=true`, you see information similar to the following displayed in the SAS log:

```
NOTE: Action 'nobs' used (Total process time):
NOTE:       real time               2.100185 seconds
NOTE:       cpu time                0.010999 seconds (0.52%)
NOTE:       total nodes             6 (192 cores)
NOTE:       total memory            1.11T
```

```
NOTE:         memory                     7.00K (0.00%)
The analytic server processed the request in 2.100185 seconds.
```

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Log
Default	false
Note	The values `true` and `false` are case sensitive.
See	CASLIB Statement
Example	In the following example, CAS server metrics information is not displayed in the SAS log: `cas.metrics=false`

cas.MODE='smp' | 'mpp'
forces a server to be started in symmetric multiprocessing mode (`smp`) or in massively parallel processing mode (`mpp`).

Valid in	casconfig_usermods.lua file
Category	Administration
Restriction	Applies to Linux only.
Interaction	cas.MODE is implicitly set to 'mpp' when cas.ELASTIC is set or cas.MACHINELIST is set and contains at least one CAS worker node or backup controller. Otherwise, cas.MODE is implicitly set to 'smp'.
Note	The server returns an error when `cas.mode='smp'` is specified for a server with a valid machine list.
Example	In the following example, the CAS server is forced to start in massively parallel processing mode (MPP). `cas.mode='mpp'`

cas.NODE='*filename*'
specifies the configuration file that is run on all CAS worker nodes.

Valid in	casconfig_usermods.lua file
Category	Server
Default	node.lua
Restrictions	Applies to Linux only.
	Any changes to node.lua should be made to node_usermods.lua.
See	"Understanding Configuration Files and Start-up Files"
Example	`cas.node='node.lua'`

cas.NWORKERS=*number*
specifies the number of worker nodes associated with this session.

Valid in	CAS statement SESSOPTS option
	casconfig_usermods.lua file
Category	Administration
Default	0
Range	0–5000
Restriction	Applies to Linux only.
Example	`cas.nworkers=8`

cas.PERMSTORE='*path*'

specifies the path to a directory where the CAS server stores permissions.

Enclose *path* in single quotation marks.

The server saves its caslib and access control information to the `cas.PERMSTORE` directory periodically and when it shuts down.

Each subsequent time that the server starts, caslib and access control information is initialized from the server's `cas.permstore` location.

IMPORTANT When you update `cas.PERMSTORE` in casconfig_usermods.lua, you must also update SASPERMSTORE in **`/opt/sas/viya/config/etc/sysconfig/cas/default/sas-cas-usermods`**. Add the line: **`export SASPERMSTORE=`*`path`***.

CAUTION

Backups of access controls are not automatically performed. In a programming-only deployment, it is strongly recommended that you periodically back up each CAS server's stored access control and caslib information. In particular, it is important to create a backup after you modify access controls or add, delete, or modify global caslibs. See *SAS Viya Administration: Backup and Restore*.

Valid in	casconfig_usermods.lua file
Category	Access Control
Restriction	Do not directly edit the files in a `cas.PERMSTORE` location.
Note	Each CAS server should have its own `cas.PERMSTORE` location. To minimize the potential for network timing issues, it is recommended that each `cas.PERMSTORE` location be on the controller machine and not on a network file system. The server creates a directory with the name of the fully qualified DNS name of the machine that the main controller is running on in the specified permstore directory. For example, if you specify `cas.PERMSTORE='/var/my_permstore'`, CAS writes its permstore to `/var/my_permstore/controller.machine.example.com`.
Examples	Here is an example defining the CAS permstore location on Linux: `cas.permstore='/opt/sas/viya/config/etc/cas/default/permstore'` Here is an example defining the CAS permstore location on Windows: `cas.permstore='C:\\ProgramData\\SAS\\Viya\\etc\\cas\\default\\permstore'`

cas.PORT=*port*

specifies the port to which the CAS server listens.

The maximum allowable port number is 65535. If a valid port is not specified, the server listens on a port selected by the operating system through the TCP/IP ephemeral port range. A common range is 32768-61000.

The commonly configured port is 5570.

Valid in	casconfig_usermods.lua file
Category	Network
See	cas.GCPORT and cas.HTTPPORT
Example	`cas.port=5570`

cas.PROVLIST='ext' | 'kerb' | 'oauth'

specifies the authentication providers that the CAS server uses to authenticate incoming user connections.

- 'ext'

  The external provider provides support for an external PAM authentication method when root access is required for authentication.

  Note: The 'ext' option applies to Linux only.

- 'kerb'

  The Kerberos provider is used only when a Kerberos ticket is provided for authentication. For more information, see “Kerberos Security” in *SAS Viya for Linux: Deployment Guide*.

- 'oauth'

  OAuth provider is always loaded (even when not listed) to support REST endpoints and communications between CAS worker nodes and the controller.

Valid in	casconfig_usermods.lua file
Category	Security
Default	oauth
Note	The CAS server configures the specified providers and uses each in order until an authenticated connection is successful.
Example	In this example, an external provider provides support for an external PAM authentication method. Although not specified, OAuth is always loaded to support REST endpoints and communications between CAS worker nodes and the controller. `cas.provlist='ext'`

cas.REMOVENODECANCELTIMEOUT='*interval*'

when quiescing sessions in preparation for moving data from nodes that are being removed, the amount of time that CAS waits for long-running actions to complete before canceling them.

Valid in	casconfig_usermods.lua file
Category	Server
Default	120 seconds

Restriction	Applies to Linux only.
Note	A value of zero indicates that a cancel request should never be sent.
Example	`cas.removeNodeCancelTimeout='600'`

cas.REMOVENODEKILLTIMEOUT='*interval*'

when quiescing sessions in preparation for moving data from nodes that are being removed, the amount of time that CAS waits for a canceled action to stop before killing its session.

Valid in	casconfig_usermods.lua file
Category	Server
Default	15 seconds
Restriction	Applies to Linux only.
Note	A value of zero indicates that the sessions should never be killed.
See	cas.REMOVENODECANCELTIMEOUT on page 82
Example	`cas.removeNodeKillTimeout='300'`

cas.RESOLVEWORKERADDRESS=true | false

specifies how CAS list node actions return CAS worker node host names.

When `true`, CAS list node actions attempt to return the list of worker node host names. If the directory name service (DNS) lookup is unresponsive, CAS cancels the lookup and `resolveworkeraddress` is automatically set to `false`.

When `false`, list node actions return only the IP address of elastically added nodes.

> **TIP** Setting cas.RESOLVEWORKERADDRESS to `false` ensures that the analytic cluster is less impacted by an unresponsive DNS configuration. However, some output is displayed as IP addresses instead of host names.

Valid in	casconfig_usermods.lua file
Category	Server
Default	True
Restriction	Applies to Linux only.
See	cas.ELASTIC
Example	`resolveworkeraddress=false`

cas.SERVICESBASEURL='*URL*'

specifies the URL that enables CAS server to authenticate and to use SAS Viya services. *URL* points to the deployed SAS Viya web services.

When set, cas.SERVICESBASEURL creates a hybrid authentication environment where username-password authentication is converted to an OAuth token and CAS can fetch groups from SAS Viya services and use features such as the credentials vault.

Valid in	casconfig_usermods.lua file

Category	Security
Notes	*URL* must match the reverse proxy server host name and port to enable CAS to communicate with SAS Viya web services. Enclose *URL* in single quotation marks.
Example	`cas.servicesbaseurl='http://company.example.com'`

cas.STARTUP='*filename*'

specifies the configuration file that the CAS server runs before the server accepts any client connections. This start-up file contains CAS actions that the server runs as it starts up.

Valid in	casconfig_usermods.lua file
Category	Server
Default	casstartup.lua
Restriction	Any changes to casstartup.lua should be made to casstartup_usermods.lua.
See	"Understanding Configuration Files and Start-up Files"
Example	`cas.startup='casstartup.lua'`

cas.STARTUPDIR='*path*'

specifies the location for the SAS Cloud Analytic Services start-up directory.

Valid in	casconfig_usermods.lua file
Category	Server
Default	**`/opt/sas/viya/config/default`**
See	"Understanding Configuration Files and Start-up Files"

Examples	Here is an example on Linux:
	`cas.startupdir='/opt/sas/viya/config/default/my_cas_startup'`
	
	Here is an example on Windows:
	`cas.startupdir='C:\ProgramData\SAS\Viya\etc\cas\default\MyCASStartup'`

cas.SUBSETSESSIONCOPIES=*number-of-blocks*

specifies the number of extra block copies made for failover in either of the following scenarios:

- a session is smaller than the full server.
- CAS reads blocks of an HDFS remotely.

Valid in	CAS statement SESSOPTS option casconfig_usermods.lua file
Category	Administration
Default	0
Restriction	Applies to Linux only.
Example	`cas.subsetsessioncopies=3`

cas.TAG='*string*'

specifies a string to name the CAS server instance that is visible in the operating system, such as in the process list.

The cas.TAG option can be useful when debugging CAS.

Valid in casconfig_usermods.lua file

Category Server

Example In this example, the string ‘cas-my_tag’ is used to name the CAS server instance:

```
cas.tag='cas-my_tag'
```

When you view the process list from a Linux command prompt, you see something similar to the following:

```
27019 ?        00:00:01  cas-my_tag
```

cas.TENANTID='*string*'

specifies the user ID for the CAS tenant. cas.TENANTID is used to validate that the authenticating user belongs to the correct CAS tenant.

Valid in casconfig_usermods.lua file

Category Administration

Restrictions Used only in multi-tenant CAS deployments.

Applies to Linux only.

Example

```
cas.tenantid='tenant1'
```

cas.TIMEOUT=*seconds*

specifies the SAS Cloud Analytic Services session time-out in seconds for a new or existing session.

Valid in CAS statement SESSOPTS option

casconfig_usermods.lua file

Category Session

Default In order of descending precedence:

1. CAS statement TIMEOUT= option value, if specified
2. SAS system option CASTIMEOUT=, if you explicitly set it in SAS to a value greater than 0
3. 60

Range 0–31536000

Notes The session time-out starts when the number of connections to the session becomes zero and no actions are executing.

If a connection is established before the time-out expires, the time-out is canceled. Otherwise, the session is automatically terminated when the time-out expires.

When set to 0, the session is terminated immediately when the connection count becomes zero.

See CASTIMEOUT= System Option

Example	`cas.timeout=100`

cas.USERLOC='%HOME' | '*pathname*/%USER'

specifies that the CAS server creates a personal caslib for each user at session start-up time in the specified location.

`'%HOME'` equates to the user's operating system `$HOME` directory.

`'`*`pathname`*`/%USER'` refers to a directory named for the user's user ID under the specified file system path. Make sure that %USER is always placed at the end of the path that is specified for the option value.

Enclose *pathname* in single quotation marks.

> **IMPORTANT** You must update `SASUSERLOCDIR` in **`/opt/sas/viya/config/etc/sysconfig/cas/default/sas-cas-usermods`**. Add the line: **`export SASUSERLOCDIR=path/%USER`**.

Valid in	casconfig_usermods.lua file
Category	Caslib
Restriction	Applies to Linux only.
Examples	In this example, the personal caslib directory is the user's operating system $HOME directory: `cas.userloc='%HOME'`
	In this example, the user's personal caslib directory is named for his or her user ID and is located under **`/local`**: `cas.userloc='/local/%USER'`

cas.USEYARN=true | false

adds a reservation request to YARN for CAS memory size.

The memory limit for the YARN request is set with cas.memorysize.

Valid in	casconfig_usermods.lua file
Category	Administration
Default	false
Restriction	Applies to Linux only.
See	cas.MEMORYSIZE on page 79
	"How to Limit Memory Use" in *SAS Cloud Analytic Services: Fundamentals*
Example	`cas.useyarn=true`

Grouped by Categories

Access Control Options

- cas.PERMSTORE on page 81

Action Options

- cas.CMPOPT on page 65

Administration Options

- cas.CPUSHARES on page 67
- cas.ELASTIC on page 71
- cas.LIFETIME on page 75
- cas.MACHINELIST on page 77
- cas.MAXCORES on page 77
- cas.MEMORYSIZE on page 79
- cas.MODE on page 80
- cas.NWORKERS on page 80
- cas.SUBSETSESSIONCOPIES on page 84
- cas.TENANTID on page 85
- cas.USEYARN on page 86

Caslib Options

- cas.COLOCATION on page 66
- cas.MAXTABLEMEM on page 78
- cas.USERLOC on page 86

Data Options

- cas.AZUREAUTHCACHELOC= on page 65
- cas.AZURETENANTID= on page 65
- cas.HDFSUSERLOC on page 72

Data Quality Options

- cas.DQLOCALE on page 70
- cas.DQSETUPLOC on page 71

DATA Step Options

- cas.DATASTEPFMTERR on page 68
- cas.DATASTEPMSGSUMLEVEL on page 68
- cas.DATASTEPREPLACETABLE on page 68
- cas.DATASTEPMERGENOBY on page 69

Formats Options

- cas.FMTSEARCH on page 71

Input Control Options

- cas.EVENTDS on page 71
- cas.INTERVALDS on page 75

Java

- cas.JREOPTIONS on page 75

Localization

- cas.LOCALE on page 76

Log Options

- cas.LOGCFGLOC on page 76
- cas.LOGFLUSHTIME on page 76
- cas.MESSAGELEVEL on page 79
- cas.METRICS on page 79

Network Options

- cas.DCHOSTNAMERESOLUTION on page 69
- cas.DCPORT on page 70
- cas.DCEXTPORT on page 70
- cas.EXTHOSTKNOWNBY on page 70
- cas.GCPORT on page 72
- cas.HOSTKNOWNBY on page 73
- cas.HTTPPORT on page 73
- cas.HTTPPORTMAX on page 73
- cas.PORT on page 82

Security Options

- cas.KEYFILE on page 75
- cas.PROVLIST on page 82
- cas.SERVICESBASEURL on page 83

Note:

Other security-related configuration file options can be found in “Configuration File Options for Data Transfer” in *Encryption in SAS Viya: Data in Motion*.

Server Options

- cas.INITIALBACKUPS on page 74
- cas.INITIALWORKERS on page 74
- cas.MAXSESSIONS on page 78
- cas.NODE on page 80
- cas.REMOVENODECANCELTIMEOUT on page 82
- cas.REMOVENODEKILLTIMEOUT on page 83
- cas.RESOLVEWORKERADDRESS on page 83
- cas.STARTUP on page 84
- cas.STARTUPDIR on page 84
- cas.TAG on page 85

Session Options

- cas.APPTAG on page 65
- cas.TIMEOUT on page 85

Sort Options

- cas.COLLATE on page 66

CAS Environment Variables

Where Do I Set CAS Environment Variables?

With one exception (see CAUTION later), you set SAS Cloud Analytic Services environment variables in the CAS controller’s configuration file, casconfig_usermods.lua. During CAS server start-up, the controller shares the configuration as each worker and the backup controller (if present) connect. By default, casconfig_usermods.lua is located in `/opt/sas/viya/config/etc/cas/default` on Linux and in `\ProgramData\SAS\Viya\etc\cas\default` on Windows.

If you want to isolate an environment variable change to a particular CAS node, then make your change in node_usermods.lua residing on the particular node machine.

TIP For sites that use Ansible, it is recommended that you make your CAS server environment variable changes to vars.yml and rerun Ansible to apply these changes. For more information, see "Modify the vars.yml File" in *SAS Viya for Linux: Deployment Guide*.

On Linux, one CAS environment variable, LD_LIBRARY_PATH, must be specified before CAS starts. Therefore, you must add this variable to the cas_usermods.settings file. If you edit cas_usermods.settings on a single node, only that node is affected. If you want to set an environment variable on every node, you must edit cas_usermods.settings on every node. By default, cas_usermods.settings is located in `/opt/sas/viya/config/etc/cas/default` on Linux.

CAUTION
SAS Cloud Analytic Services ignores any instance of LD_LIBRARY_PATH found in the server configuration file. Specify LD_LIBRARY_PATH in the cas_usermods.settings file only.

There are additional CAS configuration files and directories. For more information, see "Understanding Configuration Files and Start-up Files".

CAS Environment Variables Reference

Note: For information about SAS Cloud Analytic Services TLS environment variables, see "CAS TLS Environment Variables" in *Encryption in SAS Viya: Data in Motion*.

env.CAS_ACTION_THREAD_NICE='*niceness-priority*'
specifies the niceness priority for the CPU intensive threads that do CAS action processing.

Increasing the niceness priority value—especially during high CPU utilization—provides an opportunity for the CAS heartbeat thread to run, and prevents workers from being disconnected prematurely.

Valid in	casconfig_usermods.lua file
Category	Environment
Default	1
Range	0–19
Restrictions	`env.CAS_ACTION_THREAD_NICE` is case sensitive.
	Applies to Linux only.
See	The man page for the Linux nice command.
Example	`env.CAS_ACTION_THREAD_NICE='1'`

env.CAS_ADDITIONAL_YARN_OPTIONS='*yarn-option*…'
specifies a yarn queue or other values to provide similar flexibility to the CAS server.

Valid in	casconfig_usermods.lua file

Category	Server
Restrictions	`env.CAS_ADDITIONAL_YARN_OPTIONS` is case sensitive. Applies to Linux only.
Example	In this example, the YARN queue and priority are specified: `env.CAS_ADDITIONAL_YARN_OPTIONS='--queue sas_cas --priority 3'`

env.CAS_CFG_CONSULPATH='*path-to-kv-pairs*'

specifies the SAS Configuration Server (Consul) path to key-value pairs loaded into the CAS Options namespace and OSENV namespace, respectively. If a key does not map to a CAS server option, CAS ignores it.

Valid in	casconfig_usermods.lua file
Category	Server
Restrictions	Applies to Linux only. `env.CAS_CFG_CONSULPATH` is case sensitive.
Example	In this example, the location to the SAS Configuration Server key-value pairs for CAS configuration options and for CAS environment variables is specified: `env.CAS_CFG_CONSULPATH='/opt/sas/viya/config/data/cas/'`

env.CAS_CONTROLLER_TEMP='*path* [[:*path*] …]'

specifies the disk paths to temporarily store data on the CAS controller machine for the CAS upload action and when saving CSV files to certain cloud platforms such as Amazon S3.

Delimit multiple paths with a colon (:).

Valid in	casconfig_usermods.lua file
Category	Data
Default	When env.CAS_CONTROLLER_TEMP is not specified, CAS defaults to the locations pointed to by env.CAS_DISK_CACHE.
Restrictions	Applies to Linux only. `env.CAS_CONTROLLER_TEMP` is case sensitive. env.CAS_CONTROLLER_TEMP should point to a file system that uses only a local disk.
Example	`env.CAS_CONTROLLER_TEMP = '/upload_data/disk1:/upload_data/disk2'`

env.CAS_DISK_CACHE='*path* [[:*path*] …]'

specifies the disk paths to cache data on CAS worker machines.

On Linux, delimit multiple paths with a colon (:). On Windows, delimit multiple paths with a semicolon (;).

IMPORTANT It is a best practice to always specify `env.CAS_DISK_CACHE`. When not specified, `env.CAS_DISK_CACHE` defaults to **`/tmp`** on Linux and to the `TEMP` environment variable on Windows.

Valid in	casconfig_usermods.lua file
Category	Data
Restrictions	`env.CAS_DISK_CACHE` is case sensitive.
	`env.CAS_DISK_CACHE` should point to a file system that uses only a local disk.
Tip	There is an advantage to using multiple physical disks. When using multiple threads, mapping files can occur concurrently if multiple disks are used. Hadoop also uses this method. Therefore, there is an advantage to using a set of disks that map to both hadoop_data and CAS_DISK_CACHE directories.

Examples Here is an example on Linux:

```
env.CAS_DISK_CACHE = '/data/disk1:/data/disk2'
```

Here is an example on Windows:

```
env.CAS_DISK_CACHE = 'E:\\casDataCache;F:\\casDataCache'
```

env.CAS_QUOTA_MODE='ALLTABLES'

specifies that the quota applies on all the tables.

When this variable is not set or set to some other value, tables that are mapped only from CAS_DISK_CACHE are counted toward quota limits.

Valid in	casconfig_usermods.lua file
Category	Data
Restriction	env.CAS_QUOTA_MODE is case sensitive.

CASUSERIGNORECASE='ON'

when in effect (specified using any value), causes the CAS server to ignore the letter casing for user names during authentication, group lookup, and process launch. Always specify `CASUSERLOWERCASE` whenever specifying `CASUSERIGNORECASE`, unless instructed otherwise by SAS Technical Support.

The typical scenario for declaring `CASUSERIGNORECASE` is when users run their CAS sessions under their own host account and the user authentication system is configured to be case-insensitive and contains uppercase or mixed case user names. For more information, see "The CASHostAccountRequired Custom Group" in *SAS Viya Administration: Identity Management*.

Valid in	cas_usermods.settings file
Category	Administration
Default	off
Restrictions	Applies to Linux only.
	`CASUSERIGNORECASE` is case sensitive.
Requirement	Use with `CASUSERLOWERCASE`.
Note	To turn off `CASUSERIGNORECASE`, remove its definition.

Example In this example, CASUSERIGNORECASE is in effect:

```
export CASUSERIGNORECASE='on'
```

CASUSERLOWERCASE='ON'

when in effect (specified using any value), causes the CAS server to convert user names to lowercase during group lookup. `CASUSERLOWERCASE` is typically used in conjunction with `CASUSERIGNORECASE`.

The typical scenario for declaring `CASUSERLOWERCASE` is when users run their CAS sessions under their own host account and the user authentication system is configured to be case-insensitive and contains uppercase or mixed case user names. For more information, see "The CASHostAccountRequired Custom Group" in *SAS Viya Administration: Identity Management*.

Valid in	cas_usermods.settings file
Category	Administration
Default	off
Restrictions	Applies to Linux only.
	`CASUSERLOWERCASE` is case sensitive.
Requirement	Use with `CASUSERIGNORECASE`.
Note	To turn off `CASUSERLOWERCASE`, remove its definition.
Example	In this example, CASUSERLOWERCASE is in effect: `export CASUSERLOWERCASE='on'`

CASUSERFORCELOWER='ON'

when in effect (specified using any value), causes the CAS server to convert user names to lowercase. Using CASUSERFORCELOWER eliminates the need for either CASUSERIGNORECASE or CASUSERLOWERCASE.

The typical scenario for declaring `CASUSERFORCELOWER` is when users run their CAS sessions under their own host account and the user authentication system is configured to be case-insensitive and contains uppercase or mixed case user names. For more information, see "The CASHostAccountRequired Custom Group" in *SAS Viya Administration: Identity Management*.

Valid in	cas_usermods.settings file
Category	Administration
Default	Off
Restriction	`CASUSERFORCELOWER` is case sensitive.
Note	To turn off `CASUSERFORCELOWER`, remove its definition.
Example	In this example, CASUSERFORCELOWER is in effect: `export CASUSERFORCELOWER='on'`

env.CONSUL_HTTP_ADDR='https://*SAS-Configuration-Server-machine*:*port*'

specifies the SAS Configuration Server (Consul) machine and port used by the CAS Consul interface.

Valid in	casconfig_usermods.lua file
Category	Server
Restrictions	Applies to Linux only.

	`env.CONSUL_HTTP_ADDR` is case sensitive.
Example	`CONSUL_HTTP_ADDR='my-SAS-Configuration-Server.example.com:8501'`

env.CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT='TRUE' | 'FALSE'
when specified, enables CAS resource management policies by turning on communication with SAS Configuration Server (Consul).

Valid in	casconfig_usermods.lua file
Category	Server
Default	Off (FALSE)
Restrictions	Applies to Linux only.
	`env.CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT` is case sensitive.
Note	When env.CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT is false, the GLIBC environment variable, MALLOC_ARENA_MAX, should exceed the number of concurrent sessions.
See	For more information, see "CAS Resource Management".
Example	`env.CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT='true'`

env.CAS_ENABLE_REMOTE_SAVE=TRUE
specifies whether CAS saves blocks on remote HDFS worker nodes.

Valid in	casconfig_usermods.lua file
Category	Data
Restrictions	Applies to Linux only.
	`env.CAS_ENABLE_REMOTE_SAVE` is case sensitive.
Note	Removing `env.CAS_ENABLE_REMOTE_SAVE=true` causes CAS not to save blocks on remote HDFS worker nodes.
Example	In this example, CAS saves blocks on remote HDFS worker nodes: `env.CAS_ENABLE_REMOTE_SAVE=true`

env.CAS_HEARTBEAT_LOST_TIMEOUT='*interval*'
specifies the interval (in seconds) since the last heartbeat received from a CAS worker node before the controller treats the node as lost.

Smaller intervals detect machines that silently leave the network more quickly. Larger intervals are more tolerant of machines that might be exceptionally overloaded.

Valid in	casconfig_usermods.lua file
Category	Server
Default	120 seconds
Range	60 – (no upper limit) seconds
Restrictions	Applies to Linux only.

`env.CAS_HEARTBEAT_LOST_TIMEOUT` is case sensitive.

Example
```
CAS_HEARTBEAT_LOST_TIMEOUT='300'
```

env.CAS_INSTALL='*install-path*'
specifies the installation directory for CAS.

Valid in casconfig_usermods.lua file

Category Environment

Restriction `env.CAS_INSTALL` is case sensitive.

Examples Here is an example showing the CAS installation directory on Linux:
```
env.CAS_INSTALL='/opt/sas/viya/home/SASFoundation'
```

Here is an example showing the CAS installation directory on Windows:
```
env.CAS_INSTALL='C:\\Program Files\\SAS\\Viya\\SASFoundation'
```

env.CAS_LICENSE='*path/license-file*'
specifies the path and filename that contains the CAS license.

After CAS deployment, env.CAS_LICENSE is set to **`/opt/sas/viya/config/etc/cas/default/sas_license.txt`**. The deployment process creates a Linux symbolic link between sas_license.txt and the actual SAS license file. You must change the symbolic link whenever the name of the license file changes. For more information, see "Apply New Licenses Manually" in *SAS Viya Administration: Licensing*.

Valid in casconfig_usermods.lua file

Category Administration

Restriction `env.CAS_LICENSE` is case sensitive.

Examples Here is an example showing the location of the CAS license on Linux:
```
env.CAS_LICENSE='/opt/sas/viya/config/etc/cas/default/SASViyaV0300_09JB84_Linux_x86-64.jwt'
```

Here is an example showing the location of the CAS license on Windows:
```
env.CAS_LICENSE='C:\\ProgramData\\SAS\\Viya\\etc\\cas\\default\\SASViyaV0300_09JB84_Linux_x86-64
```

env.CAS_REMOTE_HADOOP_PATH='*SASHDAT-executables-directory*'
specifies the path to the plug-in location when CAS is using an HDFS caslib to a remote HDFS cluster.

Valid in casconfig_usermods.lua file

Category Environment

Default If not specified, defaults to **`$HADOOP_HOME/bin`**

Restrictions Applies to Linux only.

`env.CAS_REMOTE_HADOOP_PATH` is case sensitive.

Note Might be needed to accommodate a nonstandard Hadoop plug-in.

Example
```
env.CAS_REMOTE_HADOOP_PATH='$HADOOP_HOME/bin'
```

env.CAS_START_MONITOR_UI='TRUE' | 'FALSE'
specifies whether CAS Server Monitor is turned on ('TRUE') or off ('FALSE').

Valid in	casconfig_usermods.lua file
Category	Administration
Default	Off ('FALSE').
Restriction	`env.CAS_START_MONITOR_UI` is case sensitive.
Note	On programming-only deployments, `env.CAS_START_MONITOR_UI` is on ('TRUE'). On full deployments, `env.CAS_START_MONITOR_UI` is off ('FALSE').
Example	In this example, CAS Server Monitor is turned on: `env.CAS_START_MONITOR_UI='true'`

env.CAS_VIRTUAL_HOST= '*host-name*'

The external host or machine name for the controller.

Use `env.CAS_VIRTUAL_HOST` when an external HTTP client needs to use an external address that differs from the actual host name known by the operating system. A common use is when the controller machine is behind a reverse proxy server.

Valid in	casconfig_usermods.lua file
Category	Network
Restriction	`env.CAS_VIRTUAL_HOST` is case sensitive.
Example	`env.CAS_VIRTUAL_HOST='my_machine'`

env.CAS_VIRTUAL_PATH='*URL-path-suffix*'

Use this environment variable when external HTTP clients must reach the CAS controller through a reverse proxy server. This identifies the path portion of the URL for the reverse proxy.

Valid in	casconfig_usermods.lua file
Category	Network
Restriction	`env.CAS_VIRTUAL_PATH` is case sensitive.
Example	`env.CAS_VIRTUAL_PATH='/cas-qstgrd-default-http'`

env.CAS_VIRTUAL_PORT=*port*

The external port number for the controller.

Use `env.CAS_VIRTUAL_PORT` when an external HTTP client needs to use a port that differs from the actual port that is local to the controller machine. A common use is when the controller machine is behind a reverse proxy server.

Valid in	casconfig_usermods.lua file
Category	Network
Restriction	`env.CAS_VIRTUAL_PORT` is case sensitive.
Example	`env.CAS_VIRTUAL_PORT=5580`

env.CAS_VIRTUAL_PROTOCOL='http | https'

Use this environment variable when external HTTP clients must reach the CAS controller through a reverse proxy server. This identifies the protocol portion of the URL for the reverse proxy.

Valid in	casconfig_usermods.lua file
Category	Network
Restriction	`env.CAS_VIRTUAL_PROTOCOL` is case sensitive.
Example	`env.CAS_VIRTUAL_PROTOCOL='https'`

env.HADOOP_HOME='*path*'
specifies the standard HADOOP_HOME variable used by Hadoop.

Valid in	casconfig_usermods.lua file
Category	Data
Restrictions	Applies to Linux only.
	`env.HADOOP_HOME` is case sensitive.
Example	`env.HADOOP_HOME='/opt/hadoop'`

env.HADOOP_NAMENODE='*machine-name* :[*machine-name*]'
identifies which machines in the Hadoop cluster are NameNodes. There can be up to two Hadoop NameNodes. Separate machine names with a colon (:). *Machine-name* can be a name, fully qualified domain name, or an IP address for a machine.

Valid in	casconfig_usermods.lua file
Category	Data
Restrictions	Applies to Linux only.
	`env.HADOOP_NAMENODE` is case sensitive.
Example	`env.HADOOP_NAMENODE='my_namenode1:my_namenode2'`

env.SAS_RNG_METHOD='*random-number-generator*'
specifies the random number generator (RNG) used when running CAS actions and procedures.

Examples of valid RNGs are: `'PCG'` (permuted congruential generator) or `'TF2'` (Threefry, 2x64-bit counter-based). The complete set of values for *random-number-generator* is listed under the CALL STREAMINIT Routine in the *SAS Functions and CALL Routines: Reference*.

Valid in	cas_usermods.settings file
Category	Data
Default	When env.SAS_RNG_METHOD is not specified, the default RNG is **`'MTHYBRID'`** (Hybrid 1998/2002 32-bit Mersenne twister).
Example	`env.SAS_RNG_METHOD='PCG'`

env.TKTXTANIO_BINDAT_DIR='*install-path*'
specifies the installation directory for SAS linguistic binary files required to perform text analysis.

Note: This environment variable is valid only on native operating systems such as Linux.

Note: TKTGDat.sh contains the SAS linguistic binary files required to perform text analysis in SAS LASR Analytic Server with SAS Visual Analytics and to run PROC HPTMINE and HPTMSCORE with SAS Text Miner.

Valid in	casconfig_usermods.lua or cas_usermods.settings file
Category	Data
Restriction	`env.TKTXTANIO_BINDAT_DIR` is case sensitive.
Examples	Here is an example on Linux: `env.TKTXTANIO_BINDAT_DIR='/opt/sas/viya/home/SASFoundation/utilities/TKTGDat'`
	Here is an example on Windows: `env.TKTXTANIO_BINDAT_DIR='C:\\Program Files\\SAS\\Viya\\SASFoundation\\utilities\\TKTGDat'`

LD_LIBRARY_PATH=*path* :[[*path*] …]

specifies the path to search for additional shared libraries.

Separate multiple paths with a colon (:).

CAUTION

SAS Cloud Analytic Services ignores any instance of LD_LIBRARY_PATH found in the server configuration file. Specify LD_LIBRARY_PATH in the cas_usermods.setting file only. Or, if your site uses Ansible, in vars.yml.

Valid in	cas_usermods.settings file
Category	Data
Restrictions	`LD_LIBRARY_PATH` is case sensitive.
	Applies to Linux only.
Notes	Be careful when specifying `LD_LIBRARY_PATH`. The path order can affect the operation of some applications.
	The `LD_LIBRARY_PATH` export statement must be on a single line without wrapping.
Example	`export LD_LIBRARY_PATH=/var/my_libs:/share/groups_libs:$LD_LIBRARY_PATH`

env.CASDATADIR='*path-to-CAS-data-directory*'

specifies an alternate path for product caslibs instead of the default path **`/opt/sas/viya/config/data/cas/default`**.

It is recommended that this option be configured to point to a high-performance storage platform.

Note: This option can only be specified during the initial software deployment. For more information, see "Set the CAS Data Directory" in *SAS Viya for Linux: Deployment Guide*.

The CAS data directory must be accessible from the primary and secondary CAS controllers.

Valid in	cas_usermods.settings file
Category	Data

Restriction	Applies to Linux only

env.TKUTILLOC='*path*'

enables the user to specify a different directory to create temporary utility files to avoid filling up the default `/tmp` directory.

By default, the temporary files are stored in `/tmp`. To use a different directory, the administrator must set this environment variable to the desried directory before starting the CAS server.

Note: This environment variable is primarily used by the SAS Federation Server. However, this variable is also used by some CAS data feeders such as Snowflake. For more information, see "Configure Temporary Storage for SAS Utility Files" in *SAS Federation Server: Administrator's Guide*.

Valid in	casconfig_usermods.lua
Category	Data
Default	`/tmp`
Example	`export env.TKUTILLOC='myDir/UtilityFiles'`

env.CASHOMEDIRLOC='*path*'

specifies the location of the directory where each user's home directory is created and matches the user name in the system.

Valid in	casconfig_usermods.lua
Category	Environment
Restriction	Applies to Linux only.
Note	CASHOMEDIRLOC is used only if CASMAKEHOMEDIR is set.

Examples	`env.CASHOMEDIRLOC='/home'`
	Here is an example where the user name is *user01*
	`'/home/user01'`

env.CASMAKEHOMEDIR='CONTROLLER'|'WORKER''|'BACKUP'|'ALL'

creates a home directory on a CAS machine, if it is not found when the user starts a session.

- CONTROLLER

  creates a home directory on the primary controller machine only.

- WORKER

  creates a home directory on the primary controller and the worker machines.

- BACKUP

  creates a home directory on the primary controller and the backup controller.

- ALL

  creates a home directory on all the nodes.

Valid in	casconfig_usermods.lua
Category	Environment

Default	If any other value is specified, it defaults to only creating the home directory on the primary controller.
Restriction	Applies to Linux only.

env.CASHOMEDIRPERMS=*permissions*

specifies the permission mode for the home directory on a CAS machine.

Note: Permissions are used only if CASMAKEHOMEDIR is set and a home directory is created.

Valid in	casconfig_usermods.lua
Category	Environment
Default	Directory permissions are set to 0700
Restriction	■ Supported directory permission values are between 0700 and 0777 ■ Specify permissions in octal format ■ The default setting provides Read/Write/Execute permissions to the user only
Example	`env.CASHOMEDIRPERMS=750`

env.CASSTRIPOAUTH=1

strips the `@domain` that is a part of the user name when authenticating a user name and password with SASLogon.

This environment variable helps in resolving a mismatch between host authentication and LDAP authentication when host authentication requires *“user@domain”* format user names, and LDAP authentication requires *“user”* format user names without the @domain part with SASLogon.

Valid in	cas_usermods.settings
Category	Environment
Default	Does not strip the domain name that is specified after the @ character in the user name.
Example	Here is an example where the user name is *user01@company.com*. It is authenticated as: `'user01'`

env.CASUSEDEFAULTCACHE

If CASUSEDEFAULTCACHE is set, then Kerberos tickets are cached in files that are named according to the following pattern: `krb5cc`_<*UID*>_<*xxxxxx*> instead of `tkt`<*xxxxxx*> in the controller and `cc`<*xxxxxx*> on the workers.

Valid in	cas_usermods.settings
Category	Environment

env.CASUSEKERBNAME=1

When CASUSEKERBNAME is set, the Kerberos realm is included as part of the user name. Setting this environment variable is helpful for sites that store user IDs in the native Kerberos form.

Valid in	cas_usermods.settings

Category	Administration

env.SAS_EXTLANG_SETTINGS
specifies the path available to all CAS workers that contains the external languages access control XML configuration file.

Valid in	cas_usermods.settings
Category	Environment

Grouped by Categories

Administration Variables

- env.CAS_LICENSE on page 95
- env.CAS_START_MONITOR_UI on page 95
- env.CASUSEKERBNAME on page 100
- CASUSERIGNORECASE on page 92
- CASUSERLOWERCASE on page 93
- CASUSERFORCELOWER on page 93

Data Variables

- env.CAS_CONTROLLER_TEMP on page 91
- env.CASDATADIR on page 98
- env.CAS_DISK_CACHE on page 91
- env.CAS_ENABLE_REMOTE_SAVE on page 94
- env.CAS_QUOTA_MODE on page 92
- env.HADOOP_HOME on page 97
- env.HADOOP_NAMENODE on page 97
- LD_LIBRARY_PATH on page 98
- env.SAS_RNG_METHOD on page 97
- env.TKTXTANIO_BINDAT_DIR on page 97
- env.TKUTILLOC on page 99

Environment Variables

- env.CAS_INSTALL on page 95
- env.CASHOMEDIRLOC on page 99
- env.CASMAKEHOMEDIR on page 99
- env.CASHOMEDIRPERMS on page 100
- env.CASSTRIPOAUTH on page 100

- env.CASUSEDEFAULTCACHE on page 100
- env.CAS_REMOTE_HADOOP_PATH on page 95
- env.SAS_EXTLANG_SETTINGS on page 101

Network Variables

- env.CAS_VIRTUAL_HOST on page 96
- env.CAS_VIRTUAL_PATH on page 96
- env.CAS_VIRTUAL_PORT on page 96
- env.CAS_VIRTUAL_PROTOCOL on page 96

Security Variables

For information about SAS Cloud Analytic Services environment variables for:

- TLS, see "CAS TLS Environment Variables" in *Encryption in SAS Viya: Data in Motion*.
- Authentication, see "CAS_AUTH_METHOD=authinfo | kerberos" in *SAS Viya Administration: Authentication*.

Server Variables

- env.CAS_ACTION_THREAD_NICE on page 90
- env.CAS_ADDITIONAL_YARN_OPTIONS on page 90
- env.CAS_CFG_CONSULPATH on page 91
- env.CONSUL_HTTP_ADDR on page 93
- env.CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT on page 94
- env.CAS_HEARTBEAT_LOST_TIMEOUT on page 94

CAS Resource Management Policies

Overview

CAS resource management policies enable you to manage disk cache used by CAS tables through three different table categories:

Table 4 *Table Categories and Policies That Manage Them*

Table category	caslib type	caslib example	Policy
Global tables	Global	HPS, PUBLIC	`globalCaslibs`
Global tables	Personal	CASUSER, CASHDFS	`Priority-`*n*

Table category	caslib type	caslib example	Policy
Session tables[1]	Personal	CASUSER, CASHDFS	`Priority-n`

1 The sum of all tables in each session. Individual caslib limits at the session level are not supported.

TIP

With the CAS command line interface, you can create a policy template that can serve as a starting point for creating your own CAS resource management policies. For more information, see "Create Policies from JSON Templates" in *SAS Viya Administration: Using the Command-Line Interfaces*.

Global caslib Policy

Because global caslibs apply to all users, global caslibs have a unique policy.

POLICY DEFINITION

```
globalCaslibs
   _ALL_          quota
   [global-caslib  quota]
   [...]
```

global-caslib
specifies a global caslib.

quota
specifies the maximum amount of CAS_DISK_CACHE space, in bytes, that *global-caslib* can use.

ALL
specifies all global caslibs.

When `_ALL_` is specified, *quota* specifies the total amount of CAS_DISK_CACHE space that can be used for all global tables, regardless of the caslibs to which the tables belong.

Example

```
  globalCaslibs
     _ALL_        400000000
     HPS          200000000
     MyGlobal     100000000

  [CAS-server-priority-n
     cpu                - share
     globalCasuser      - quota
     globalCasuserHdfs  - quota
     sessionTables      - quota]
  [...]

  priorityAssignments
     user    priority-level
     [...]
```

In this example, the shared global caslibs HPS and MySiteGlobal can use up to 200GB and 100GB of CAS_DISK_CACHE space, respectively. The maximum amount of CAS_DISK_CACHE space that all global caslibs can use is 400GB.

Priority-Level Policies

Priority-level policies assign CAS_DISK_CACHE space quotas and CPU utilization shares based on a user's group membership, or a user's explicit assignment with the `priority-assignment` option. You can define up to five priority-level policies.

POLICY DEFINITION

```
[CAS-server-priority-n
   cpu               - share
   globalCasuser     - quota
   globalCasuserHdfs - quota
   sessionTables     - quota]

[...]
```

CAS-server
specifies a CAS server (for example, cas-shared-default or cas-tenant1-default.)

priority-*n*
specifies the priority level for a policy. *n* is a number 1–5.

cpu *share*
specifies the maximum amount of the CPU's capacity that all users in the associated resource management group have access to. *share* is a number that is relative to the sum of all the shares used in a CAS server's policies. The recommended practice is to define shares for all policies to total 100—thus defining percentages.

quota
specifies the maximum amount of CAS_DISK_CACHE space, in bytes, that the associated table category can use.

globalCasuser
specifies the quota for global tables that are defined in a user's personal caslib (the CASUSER caslib).

globalCasuserHdfs
specifies the quota for global tables that are defined in a user's personal caslib used on distributed server file systems such as DNFS and HDFS. (This is the CASUSERHDFS caslib.) CASUSERHDFS is created by default in MPP (massively parallel processing) mode, but can be disabled by setting the collocation to *none*.

sessionTables
specifies the quota to be placed on session tables (for example, MyTable).

Example

```
globalCaslibs
   _ALL_          quota
   [global-caslib  quota]
   [...]

  priorityLevels
    cas-shared-default-priority-1
      cpu               - 50
      globalCasuser     - 500000000
      globalCasuserHdfs - 500000000
      sessionTables     - 500000000
    cas-shared-default-priority-2
      cpu               - 20
```

```
        globalCasuser     - 50000000
        globalCasuserHdfs - 50000000
        sessionTables     - 50000000
     cas-shared-default-priority-3
        cpu               - 30
        globalCasuser     - 10000000
        globalCasuserHdfs - 10000000
        sessionTables     - 10000000

  priorityAssignments
     user    priority-level
     [...]
```

In this example, three out of a maximum of five policies are defined for the CAS server, cas-shared-default. The policy, `cas-shared-default-priority-1`, applies to CAS users who are members of the user group with the same name, or to CAS users who are explicitly assigned with the `priority-assignment` option. CAS users for which the `cas-shared-default-priority-1` applies to can use up to 500GB of disk cache space in their personal caslibs for shared global caslibs and shared global caslibs across a distributed server file system (such as DNFS or HDFS). These CAS users can use up to 500GB disk cache space for session tables. CAS sessions running under the `cas-shared-default-priority-1` policy are limited to a 50% share of the CAS server's CPU capacity.

Priority Assignments

Users that are not a member of a resource management user group can be explicitly assigned to a priority-level policy.

POLICY DEFINITION

```
  priorityAssignments
     user     priority-level
     [...]
```

user

specifies a CAS user ID that is not a member of a CAS resource management user group (for example, `cas-shared-default-priority-1`).

TIP Instead of a user ID, *user* can be an asterisk (*), that includes any user IDs that do not match any of the specified user IDs or CAS resource management group names.

priority *priority-level*

specifies a priority level that maps to a specific policy. *priority-level* is a number 1–5.

For example, if 4 is specified, *user* is assigned to the `cas-shared-default-priority-4` policy.

IMPORTANT If a priority level is specified for which there is no corresponding policy defined, the user has no policy assignment.

Example

```
  globalCaslibs
     _ALL_         quota
     [global-caslib  quota]
     [...]

  [CAS-server-priority-n
```

```
         cpu               - share
         globalCasuser     - quota
         globalCasuserHdfs - quota
         sessionTables     - quota]
      [...]

   priorityAssignments
      userA     1
```

In this example, three out of a maximum of five policies are defined for the CAS server, cas-shared-default. UserA is not a member of any of the CAS resource management user groups (for example, `cas-shared-default-priority-1`). Therefore, when UserA starts a CAS session, the `cas-shared-default-priority-1` policy contains the resource definitions.

See Also

"CAS Resource Management"

CAS Configuration and Start-up Logging

How Do I Use Configuration and Start-up Logging?

In addition to its normal server logging, CAS can also log when it processes its configuration and start-up files. This logging feature is similar to the SAS 9 Logging Facility, `log4sas`, and uses the `App.cas.config` logger. When you customize an existing usermods Lua script or write your own Lua script, you can include logging functions to write records to the standard CAS server log files in **`/var/log/sas/viya/cas/default/`** or to custom log files that you define.

Note: For information about managing CAS logging after configuration and start-up file processing, see "Logging: How To" in *SAS Viya Administration: Logging*.

CAS writes configuration and start-up logging records with the following log functions (documented later):

- `log.level='`*`level`*`'`
- `log.file='`*`custom-log-filename`*`'`
- `log.`*`level`*`('`*`custom-message`*`')`

You can add the log functions in the following locations:

- casconfig_usermods.lua
- casstartup_usermods.lua
- Lua scripts in the **`conf.d`** directory
- Lua scripts in the **`start.d`** directory

For more information, see "Standard Configuration Files".

Configuration and Start-up Logging Reference

log.level='[trace | debug | info | warn | error | fatal] | [1 | 2 | 3 | 4 | 5 | 6]'

specifies the lowest logging level for configuration and start-up file logging to write messages in the log file during CAS configuration and start-up file processing. The lowest level is `trace` or `1` (most verbose). The highest level is `fatal` or `6` (least verbose).

> **IMPORTANT** The `log.level` function only impacts CAS configuration and start-up logging and does not impact standard CAS server logging.

Default log.level='trace'

Restriction `log.level` is case sensitive.

Examples In this example, log messages that are at the WARN (4) level and higher are written to the log file:

```
log.level='4'
```

In this example, **`log.level`** is used to obtain the current logging level and as an input to the **`log.all('`*`custom-message`*`')`** function. The Lua string concatenation operator, '..' (two dots) is also used:

```
log.all('The current log level is: '..log.level)
```

The following text is written to the log:

```
The current log level is: ERROR
```

log.file='*custom-log-filename*' | '+*custom-log-filename*'

specifies the absolute path and custom log filename in which to write log messages during CAS configuration and start-up file processing. Using a plus sign (+) means that the log messages are appended to an existing log file.

> **IMPORTANT** The `log.file` function only impacts CAS configuration and start-up logging and does not impact standard CAS server logging.

> **TIP** You can use various Lua functions in the `log.file` and `log.`*`level`*`('`*`custom-message`*`')` functions. Also, the log functions have built-in tags that, when used, write the CAS server process ID (%P) and the process owner ID (%U). For more information, see the examples that follow and https://www.lua.org/pil/22.1.html.

Restriction `log.file` is case sensitive.

Note When `log.file` is not used, CAS writes configuration and start-up logging to the CAS server log file in `/var/log/sas/viya/cas/default/`.

Examples In this example, CAS configuration and start-up file logging is appended to a pre-existing log file, as indicated with a plus sign (+):

```
log.file='+/var/log/sas/viya/cas/default/config_startup.log'
```

In this example, CAS configuration and start-up file logging is written to a unique file that contains the current date and the CAS server process ID. The date portion of the log filename is obtained using the Lua **`os.date`** function with date tags (**`%Y`**, **`%m`**, **`%d`**). The CAS server process ID is obtained with a built-in **`log.file`** function tag, **`%P`**. The date and process ID are joined using the Lua string concatenation operator, '..' (two dots):

```
log.file=os.date('+/var/log/sas/viya/cas/default/%Y-%m-%d_controller-1_') .. '_%P.log'
```

Here is an example of the resulting log filename: **`2018-06-21_controller-1_11634.log`**

log.[all | trace | debug | info | warn | error | fatal]('*custom-message*')

writes a custom message (a string) to the log file at the specified level during CAS configuration and start-up file processing. When `ALL` is specified, the current logging level is ignored and the custom message is always written to the log file.

TIP You can use various Lua functions in the `log.`*`level`*`('`*`custom-message`*`')` and `log.file` functions. Also, the log functions have built-in tags that, when used, write the CAS server process ID (%P) and the process owner ID (%U). For more information, see the examples that follow and https://www.lua.org/pil/22.1.html.

Restriction `log.`*`level`* is case sensitive.

Examples In this example, the following custom message is written to the log file when the current logging level is set to INFO (3) or lower:

```
log.info('CAS writes this message when the current log level is INFO or lower.')
```

Therefore, any higher level custom messages (WARN, ERROR, and FATAL) are also written to the log. But, any lower level custom messages (TRACE or DEBUG) are not written to the log.

In this example, the following custom message is written to the log, using the Lua **`os.time`** function to obtain the time at which the custom message is logged:

```
log.debug(os.time('Debug message at time: %X'))
```

Here is an example of what is written to the log:

```
Debug message at time: 14:22:38
```

In this example, the following custom message is always written, regardless of the log level:

```
log.all('This is a message written at any logging level.')
```

gridmon.sh Commands

Overview

This section describes commands that you can use to operate gridmon.sh. For usage information, see “Use gridmon.sh (Linux)”.

This section is organized into these sub-sections:

- “Global Commands ”

- "Job Mode Commands "
- "Machine Mode Commands "
- "Disk Mode Commands "
- "Show Ranks Menu Commands "
- "Show Details Menu Commands "
- "Details Menu Commands "

Global Commands

Note: Menu options that produce lengthy results redirect the output to your vi editor. Closing vi returns to gridmon.

Table 5 Global Commands

Command	Description
q	Exits gridmon.sh.
Up and Down arrows Page Up and Page Down	Moves through the list of jobs, machines, or disks.
Backspace Escape	Cancels current menu, prompt, or sub-mode.
?	Shows help information for gridmon.sh.

For usage information, see "Use gridmon.sh (Linux)".

Job Mode Commands

Table 6 Job Mode Commands

Command	Description
j	Runs gridmon.sh in job mode.
Left and Right arrows	Changes the column for sorting the list.
h Home	Moves to the top of the list.
Enter	Shows the menu option for the selected job.

For usage information, see "Use gridmon.sh (Linux)".

Machine Mode Commands

Table 7 Machine Mode Commands

Command	Description
m	Runs gridmon.sh in machine mode.
Enter	Shows menu options for the selected machine

For usage information, see "Use gridmon.sh (Linux)".

Disk Mode Commands

Table 8 Disk Mode Commands

Command	Description
d	Runs gridmon.sh in disk mode.
Enter	Shows selected disk use on machines where the disk is present.

For usage information, see "Use gridmon.sh (Linux)".

Show Ranks Menu Commands

When gridmon.sh is in job mode, you display the **Show Ranks** menu by pressing `Enter` from the main window.

Table 9 Show Ranks Menu Commands

Command	Description
Show Ranks	Displays all the ranks belonging to the job and the machines on which they are running.
Kill job	Kill the selected job.
Kill jobs with user: *user-ID*	Kills all jobs of the selected user.
Kill jobs with user: *user-ID* ID: *process-ID*	Kills all jobs of the selected user and specific ID.
Kill jobs at least this old	Kills all jobs at least as old as the selected job.

Command	Description
Stack Trace all Ranks	Runs the gstack application on all processes in this job and collects results. gstack displays its results in your vi editor.

For usage information, see “Use gridmon.sh (Linux)”.

Show Details Menu Commands

When gridmon.sh is in job mode, you display the **Show Details** menu when you press `Enter` from the Ranks window (**Enter** ⇨ **Show Ranks**).

Table 10 Show Details Menu Commands

Command	Description
Show Details	Shows process ID, CPU use, virtual memory, and if not zero, the following fields: ■ **DFSSize**: Disk space in CAS_DISK_CACHE owned by the current process. ■ **HDFSSize**: Disk space mapped from HDFS. ■ **DNFSSize**: Disk space mapped from DNFS. ■ **Global FSSize**: Disk space in CAS_DISK_CACHE for global tables, owned by the main server process. ■ **CGroup Limit**: Size of memory cgroup, as specified by cas.MEMORYSIZE. ■ **CGroup Usage**: Amount of the CGroup memory that is in use by all processes belonging to this server on the current machine. ■ **Faults/s**: The number of page faults per second for the process, most commonly caused by paging in table data. (Faults can help you determine whether the process is paging.)
Kill Rank	Kills the selected rank or process.
Stack Trace	Runs the gstack application on all processes in this job and collects results. gstack displays its results in your vi editor.
Process Limits	Displays the contents of `/proc/`*`pid`*`/limits`.
FileHandle Count	Counts the files owned by the process.
FileHandle List	Lists the files owned by the process.
Environment	Displays the process’s environment handles from `/proc/`*`pid`*`/environ`.
List Memory Maps	Shows the process’s memory maps from `/proc/`*`pid`*`/maps`.
Numa Stats	Shows the output from the Linux `numastat` command for this process.

Command	Description
Show CGroups	Shows the Linux cgroups that this process belongs to.
Xterm[1]	Starts an Xterm on the selected machine.
Perf Top[1]	Runs the perf top application on this process in a new Xterm window. **Note:** The perf package must be installed.
Attach Debugger[1]	Attaches a debugger to the running process. Requires a new X window. **Note:** Attach Debugger is for use only when directed by SAS Technical Support or by SAS R&D.

1 Requires that an X Server be running on the CAS controller machine.

For usage information, see "Use gridmon.sh (Linux)".

Details Menu Commands

When gridmon.sh is in machine mode, you display the **Details** menu by pressing `Enter` from the main window.

Table 11 Details Menu Commands

Command	Description
Details	Displays information about the machine such as CPU utilization, free memory, and total memory.
Top	Runs the top application on all processes in this job and collects results. Top displays its results in your vi editor.
Xterm[1]	Starts an Xterm on the selected machine.
Perf Top[1]	Runs the perf top application on this process in a new Xterm window.

1 Requires that an X Server be running on the CAS controller machine.

For usage information, see "Use gridmon.sh (Linux)".

SAS Cloud Analytic Services: Troubleshooting

Failed to open temporary file for upload (80BFE801): /tmp/cascache1/ _f_43d6c87c_7f5d854996e8.sas7bdat

Explanation:

Insufficient disk space in CAS_DISK_CACHE on the CAS controller.

Resolution:

Add more disk space.

The first data quality operation that is performed after CAS starts takes longer than normal to execute.

Explanation:

Data quality operations require access to the CAS table containing the QKB on all workers in the analytics cluster. Loading this table takes longer for the first data quality operation after CAS starts because the table has not been loaded before. (Subsequent loads take less time.) CAS automatically loads this table when it starts. However, CAS tables for all non-default QKBs are not loaded automatically.

Resolution:

If CAS sessions use non-default QKBs, load these non-default QKBs as part of starting CAS. In the CAS configuration file, casstartup_usermods.lua, add a qkb.loadQKB() action for each non-default QKB. For more information, see "Understanding Configuration Files and Start-up Files".

From SAS Studio 4.4, CAS Server Monitor is inaccessible

Explanation:

In full SAS Viya deployments, CAS Server Monitor is turned off by default and cannot be launched from the SAS Studio 4.4 banner.

Resolution:

Use another interface to perform CAS administration such as SAS Environment Manager or the SAS Viya command line interface.

A CAS server with process id <*process-id*> is currently running and has exclusive access to the access control storage location <*storage-location*>

Explanation:

This error message is displayed when a CAS server fails to access the permstore and terminates because the permstore is already being accessed by another CAS server.

On Windows, the error message is as follows: `A CAS server is currently running and has exclusive access to the access control storage location <storage-location>.`

SAS Cloud Analytic Services: Interfaces

There are several interfaces that you can use to administer a CAS server. The following table lists these interfaces and the shading indicates the relative amount of CAS administration that each covers:

Table 12 Interfaces to CAS Administration

◕	CAS Server Properties action set	A programmatic interface for CASL (the CAS procedure), Python, Lua, and R. Used to display server option values.

◔	Ansible	A software orchestration tool that provides a straightforward approach to deploying and provisioning SAS Viya.
◒	Administrative scripts	Scripts used to operate CAS server, change the process owner account, and to convert from single- to multi-machine CAS.
◕	Command-line interface	A command-line interface that enables you to perform CAS administration.
◕	SAS Environment Manager	A graphical enterprise web application used to modify and view a subset of server properties and to adjust caslib management privileges.
●	CAS Server Monitor	A graphical web application that is embedded in the CAS server. Used to view server information and to manage sessions, nodes, and caslib management privileges.